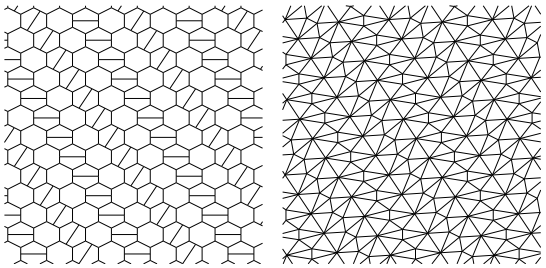
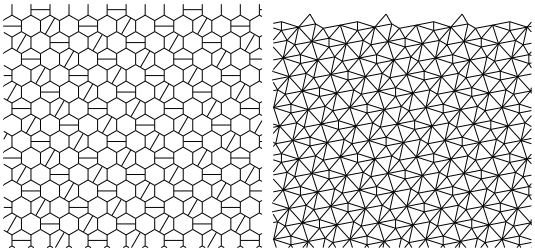
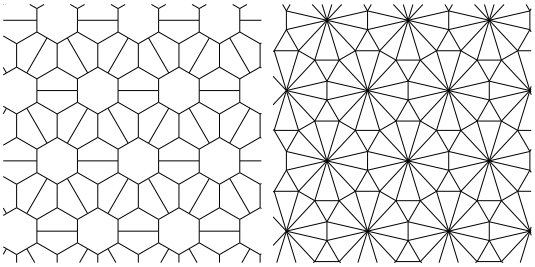


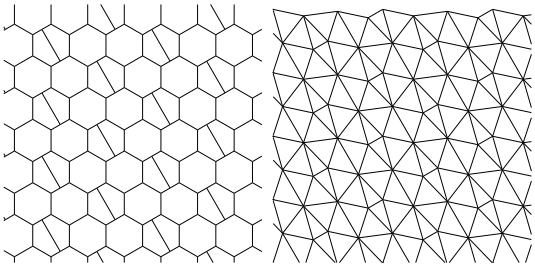
(20) $5_3[3^5]8_6[3^8]_{\mathrm{II}}$



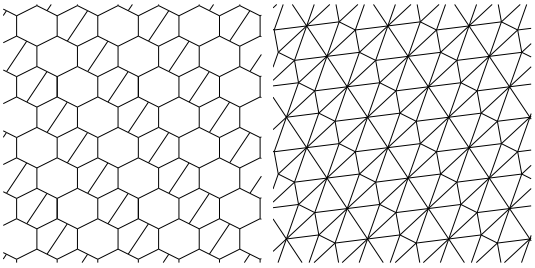
(21) $5_3[3^5]8_6[3^8]_{\mathrm{III}}$



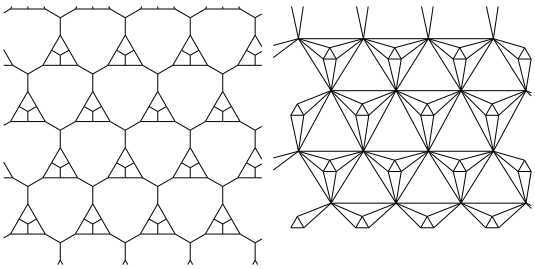
(22) $5_2[3^5]12_{12}[3^{12}]$



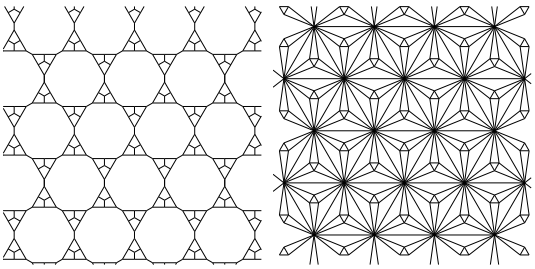
(23) $5_4[3^5]7_4[3^7]_{\mathrm{II}}$



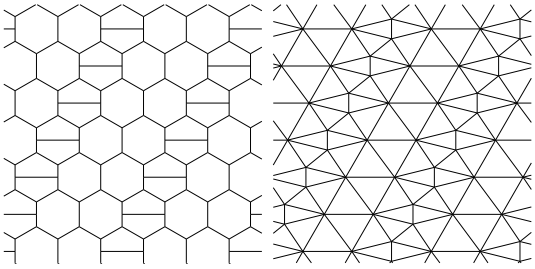
(24) $5_3[3^5]8_6[3^8]_{\mathrm{I}}$



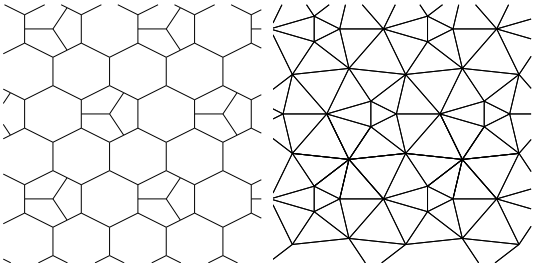
(25) $4_2[3^4]12_6[3^{12}]$



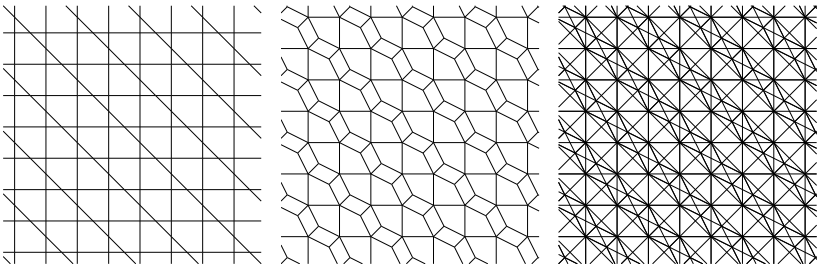
(26) $4_2[3^4]18_{12}[3^{18}]$



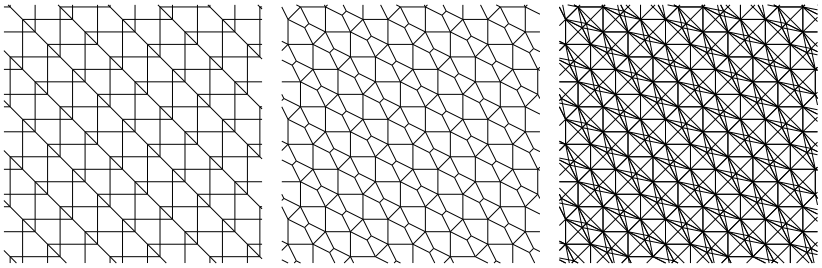
(27) $5_3[3^5]7_3[3^7]_{\mathrm{I}}$



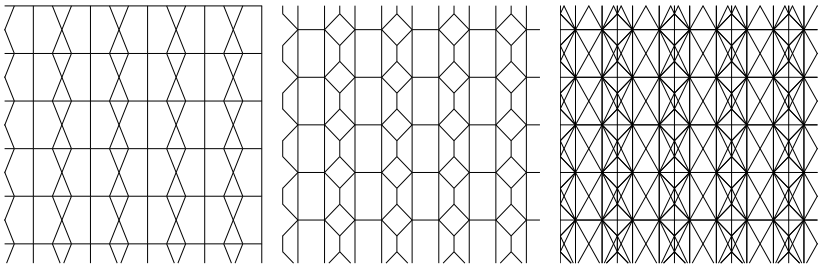
(28) $5_3[3^5]7_3[3^7]_{\mathrm{II}}$



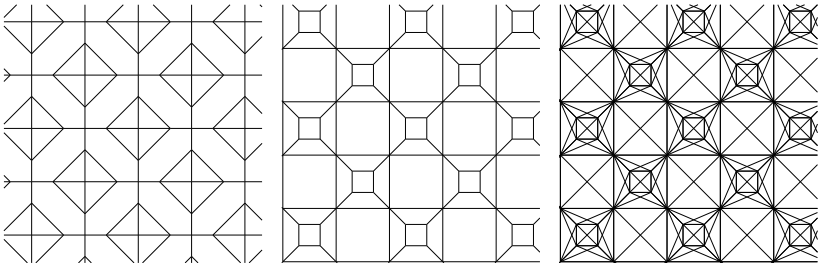
(29) $3_2[4^3]5_2[4^5]_{\mathrm{I}}$



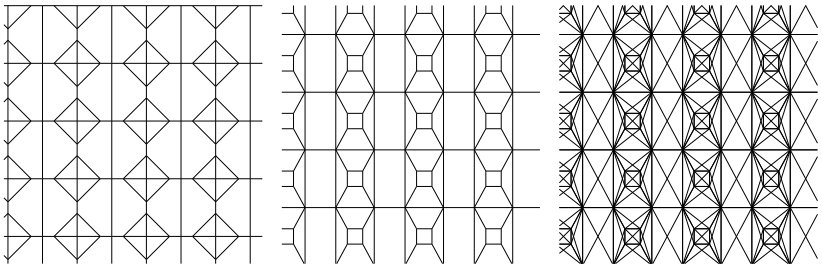
(30) $3_2[4^3]5_2[4^5]_{\mathrm{II}}$



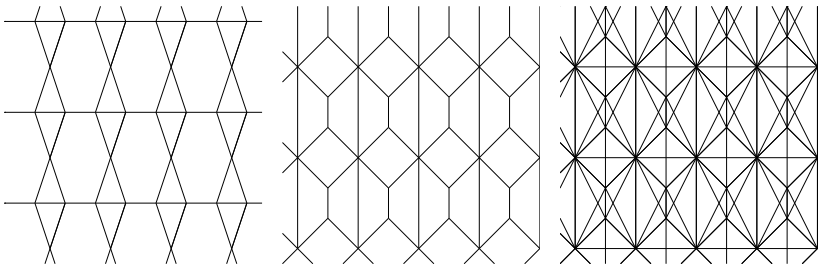
(31) $3_3[4^3]5_3[4^5]$



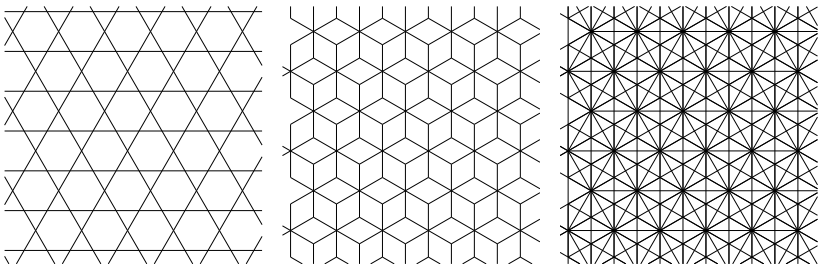
(32) $3_1[4^3]6_2[4^6]_{\mathrm{I}}$



(33) $3_1[4^3]6_2[4^6]_{\mathrm{II}}$



(34) $3_2[4^3]6_4[4^6]$



(35) $3_3[4^3]6_6[4^6]$

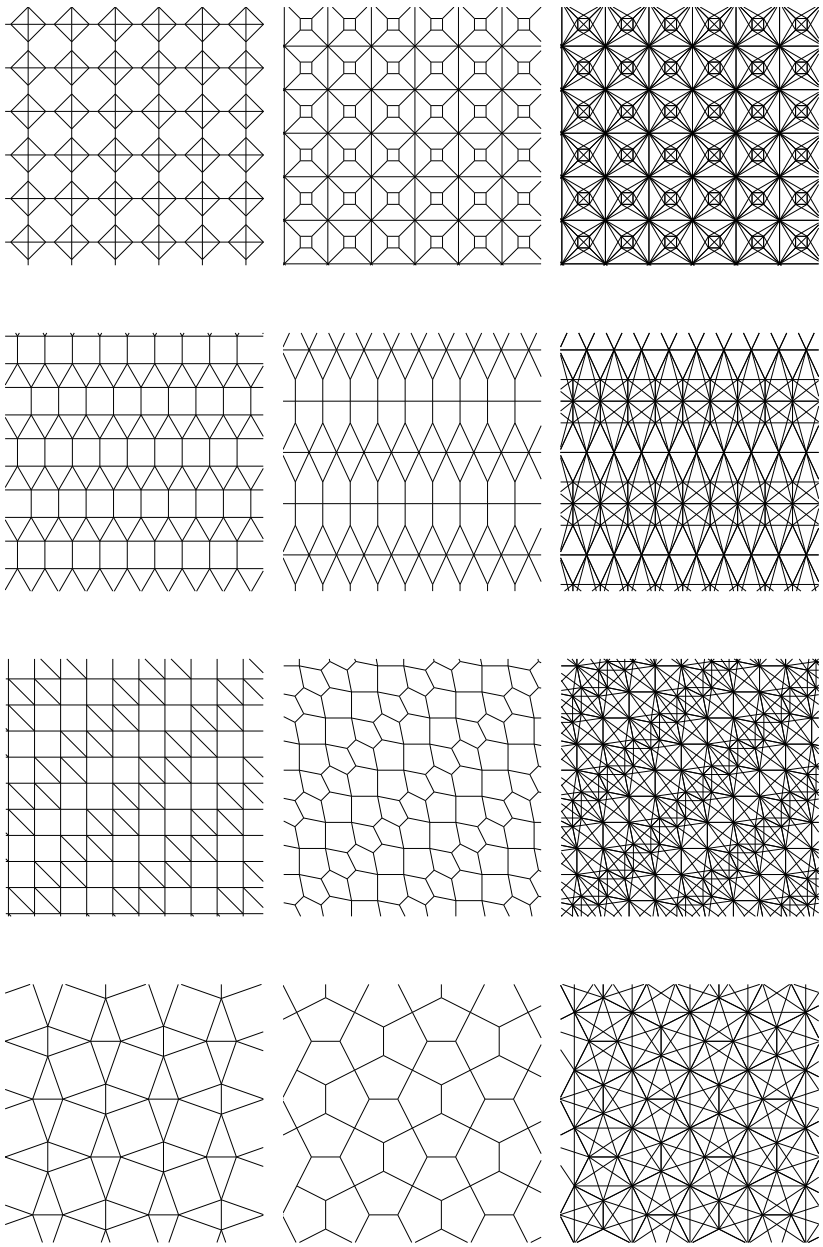


Figure 4.12 *The thirty-nine 2-homohedral tilings.*

Once while doing the above simulations I came across an interesting batch of the $4_4[3^4]7_2[3^7]_{\mathrm{II}}$ 2-homohedral tiling where the percolations forward and backward in time gave the same value of p_c, that is there is an approximate time symmetry. In other words the p_c's of both the first and the second population are symmetric to each other. Each percolates when its population reaches 756 out of the combined total of 1070, which identically gives $p_c = 0.7065$. Large non-percolating clusters quickly disappears after the onset of percolation as shown in Figure 4.13. In Figure 4.13 (b) the size of the second largest cluster is reduced to one between the population II of 941 and 961, which correspond to the density of 0.8794 and 0.8981 respectively. Interestingly from the population of 961 until 1070 there is only one cluster despite the fact that there are still more than one hundred population I. Here *population II* always refers to any population which percolates or which is being observed with regard to percolation. On the other hand in Figure 4.13 (b) the second biggest cluster is the smallest cluster from population 898 to 1037. Again from the population of 1038 until the maximum 1070 no second clusters appear.

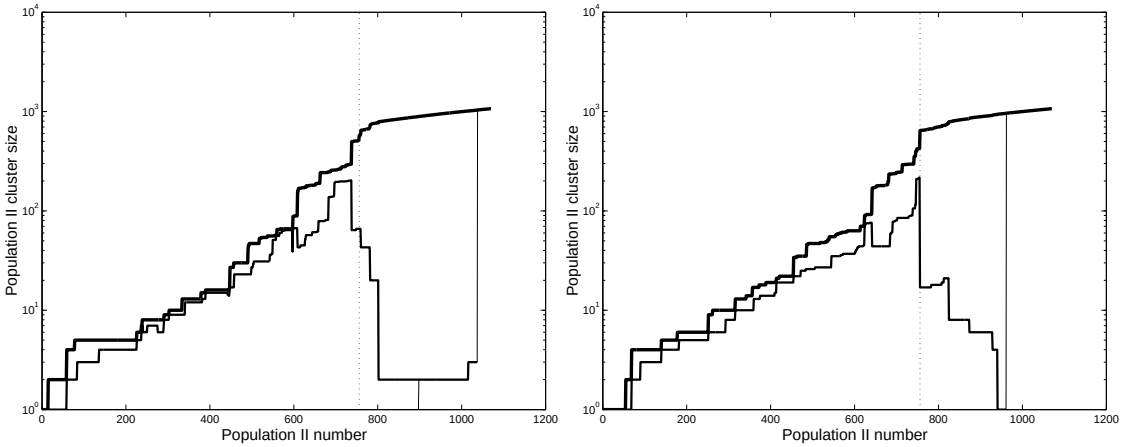
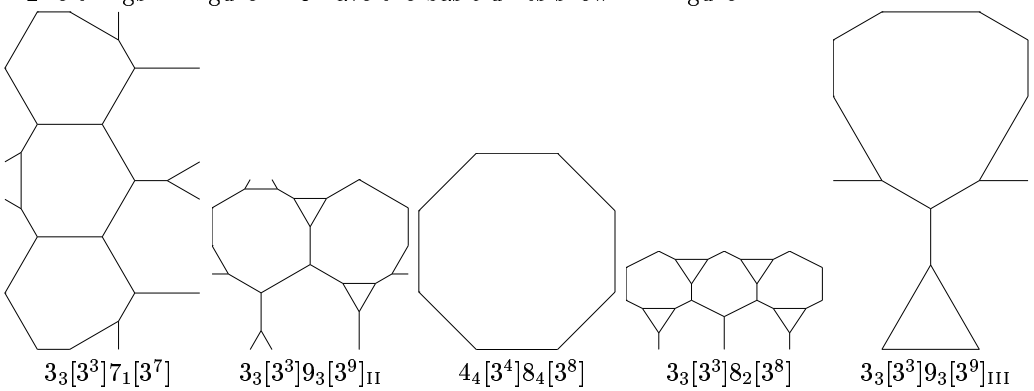
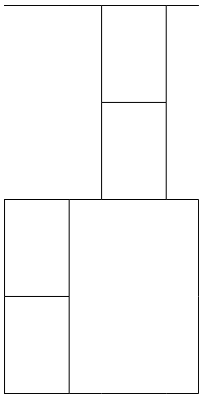


Figure 4.13 *The biggest, the second biggest, and the smallest cluster from a batch of the* $4_4[3^4]7_2[3^7]_{\mathrm{II}}$ *2-homohedral tiling where the percolation happens to be symmetry in time, that is (a) and (b) are time symmetric to each other. The vertical dotted line is where the percolation point occurs in each case. The the heaviest to the lightest lines are respectively the progression of the clusters which are biggest, second biggest, and smallest at each population density.*

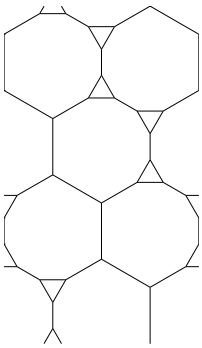
The percolation considered is that which percolates from the lower bound to the upper bound, both of which are fixed. Because the networks must be constructed such that there would be duplicates of neither vertices nor links, there must be cells some vertices and edges of which belong to one or more of the basic tiles adjacent to them. As a result, those basic tiles on the boundary sometimes have a few of their cells missing, and depending on whether the basic tile in question is very complex or simple the position of the missing cell may merely lie on its boundary or may lie nearly halfway into its interior as is the case with the $4_3[3^4]8_3[3^8]_{\mathrm{I}}$ and the $4_3[3^4]8_3[3^8]_{\mathrm{III}}$ tilings. The upper and the lower boundaries, which decide the percolation, are chosen as straight lines parallel to the vertical axis and lying in the direction towards the interior of the network and at a fixed distance from the the maximum and the minimum horizontal positions respectively of all the vertices in the network. This fixed distance is defined for each network to be a fixed ratio of its overall width. The value of 0.05 had been used which was later changed to 0.1 due to a problem which occured while simulating the $4_2[3^4]8_2[3^8]$ tessellation, missing cells formed a continuous band disconnecting all the cells on the boundary from the rest of the network, the upper boundary lay within this band when shifted by five percent from the upper limit but not when shifted by ten percent.

The tilings in Figure 4.13 have the basic units shown in Figure 4.14.

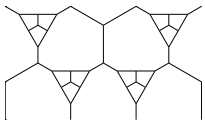




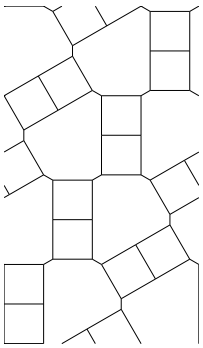
$4_3[3^4]10_6[3^{10}]_{\mathrm{I}}$



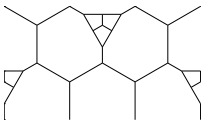
$3_3[3^3]9_3[3^9]_{\mathrm{I}}$



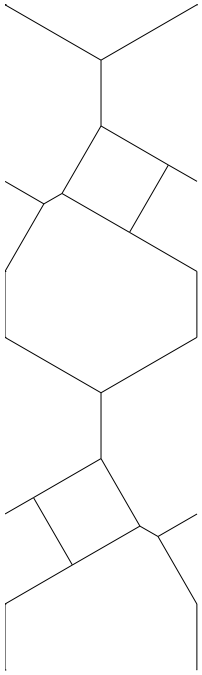
$4_2[3^4]10_4[3^{10}]$



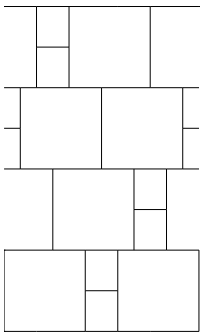
$4_3[3^4]10_6[3^{10}]_{\mathrm{II}}$



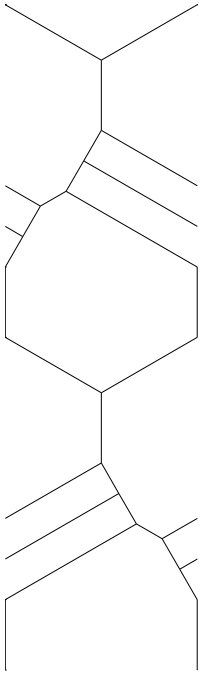
$4_2[3^4]8_2[3^8]$



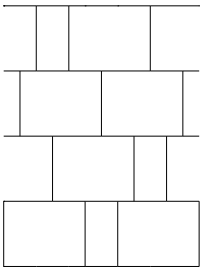
$4_3[3^4]8_3[3^8]_{\mathrm{I}}$



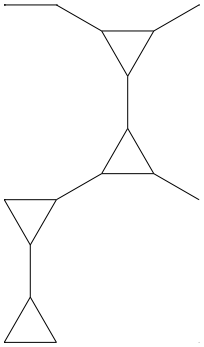
$4_3[3^4]8_3[3^8]_{\mathrm{II}}$



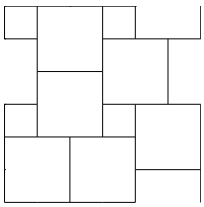
$4_3[3^4]8_3[3^8]_{\mathrm{III}}$



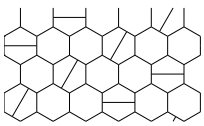
$4_4[3^4]7_2[3^7]_{\mathrm{II}}$



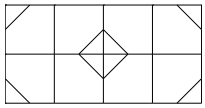
$3_3[3^3]12_6[3^{12}]$



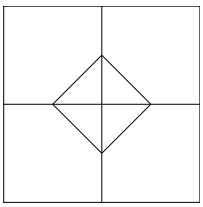
$4_4[3^4]7_2[3^7]_{\mathrm{I}}$



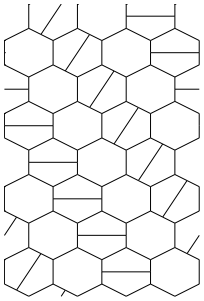
$5_4[3^5]7_4[3^7]_{\mathrm{I}}$



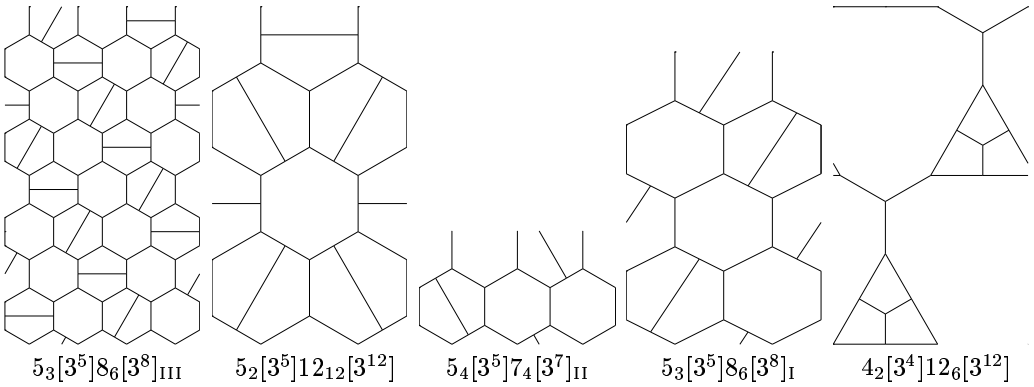
$3_1[4^3]5_1[4^5]_{\mathrm{I}}$



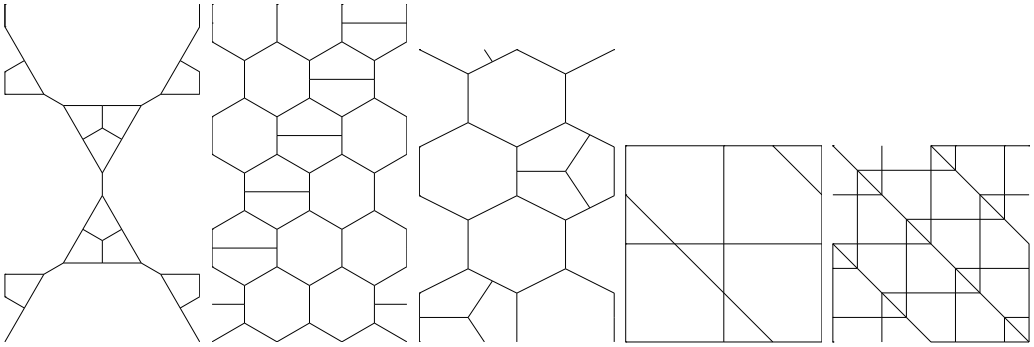
$3_1[4^3]5_1[4^5]_{\mathrm{II}}$



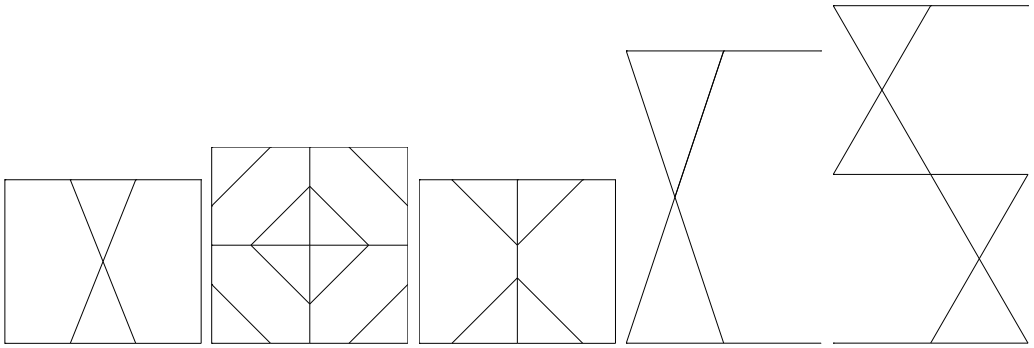
$5_3[3^5]8_6[3^8]_{\mathrm{II}}$



$5_3[3^5]8_6[3^8]_{III}$ $5_2[3^5]12_{12}[3^{12}]$ $5_4[3^5]7_4[3^7]_{II}$ $5_3[3^5]8_6[3^8]_I$ $4_2[3^4]12_6[3^{12}]$



$4_2[3^4]18_{12}[3^{18}]$ $5_3[3^5]7_3[3^7]_I$ $5_3[3^5]7_3[3^7]_{II}$ $3_2[4^3]5_2[4^5]_I$ $3_2[4^3]5_2[4^5]_{II}$



$3_3[4^3]5_3[4^5]$ $3_1[4^3]6_2[4^6]_I$ $3_1[4^3]6_2[4^6]_{II}$ $3_2[4^3]6_4[4^6]$ $3_3[4^3]6_6[4^6]$

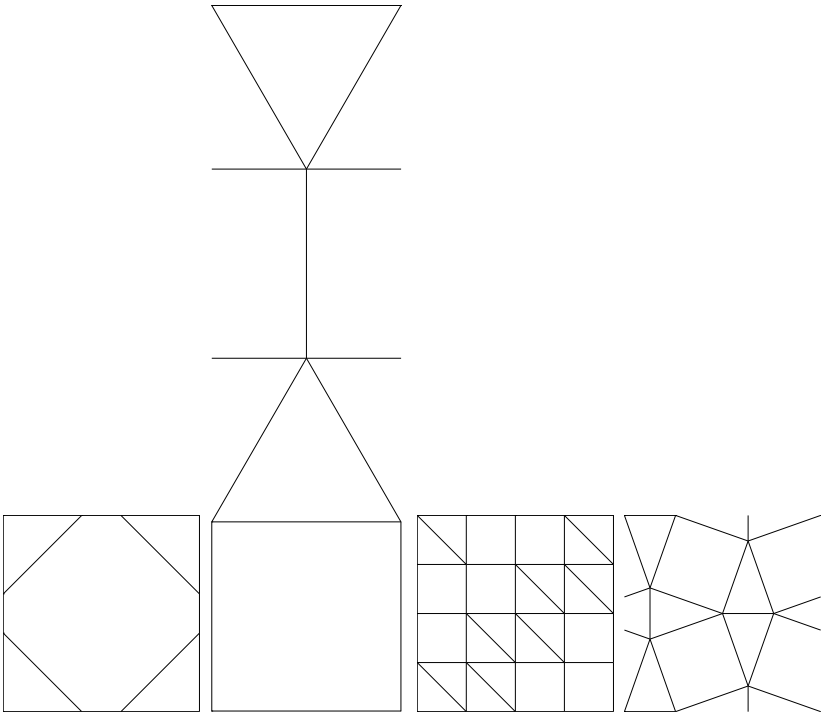


Figure 4.14 *The unit cells of the* 2*-homohedral tilings.*

In Table 4.7 there are six types of percolation probabilities. Two cells can be neighbours to each other under the criterion of having either one or two vertices in common. This gives rise to two different values of p_c's for cells, and another two for bonds. These can be termed 1-neighbours and 2-neighbours respectively. In Physics and other physical sciences 2-neighbours are more important because the greater the contact area is the stronger the bonding. This is not always the case in other fields, for example in Sociology a smaller point of contact sometimes implies the less duplications of neighbours' neighbours, and thus the larger the resultant networks. Because there is a cost for maintaining the connections, the smaller each contact point is and the larger the network the better, therefore here 1-neighbours seem to be more important.

$3_3[3^3]7_1[3^7]$

Cell	Bond	cell	bond	vertex	edge
0.4714 ± 0.0667 1.1551×10^{-6} 4.0885×10^{-5}	0.3041 ± 0.0272 -3.3740×10^{-6} 1.1316×10^{-6}			0.7212 ± 0.0444 -1.1124×10^{-4} 1.9314×10^{-5}	0.6771 ± 0.0381 -5.2007×10^{-5} 9.5566×10^{-6}
5[99], 4[195], 5.2525 5.4667 12[391] 5.6215	12[1099] 10.4149			4[256], 4[470], 2.7266 2.7957 12[894] 2.8523	4[349], 4[657], 3.7077 3.7869 12[1275] 3.8494

$3_3[3^3]9_3[3^9]_{\mathrm{II}}$

Cell	Bond	cell	bond	vertex	edge
0.4966 ± 0.0655 -8.2288×10^{-5} 3.7765×10^{-5}	0.2734 ± 0.0206 -5.5513×10^{-6} 5.9281×10^{-7}	0.5042 ± 0.0574 -1.0585×10^{-4} 2.1108×10^{-5}	0.3119 ± 0.0417 4.8211×10^{-5} 7.5982×10^{-6}	0.7628 ± 0.0228 1.0042×10^{-6} 7.3299×10^{-7}	0.7094 ± 0.0435 -2.7590×10^{-5} 7.7345×10^{-6}
4[176], 2[139], 5.2727 5.1942 2[305], 6[412] 5.4492 5.5243	6[831], 6[1138] 12.0144 12.1564	2[305], 6[412] 5.0557 5.1165	2[771], 6[1054] 11.0298 11.1556	4[415], 2[334], 2.8193 2.7964 2[692], 6[919] 2.8584 2.8770	4[585], 2[467], 3.8120 3.7859 2[989], 6[1322] 3.8544 3.8744

$4_4[3^4]8_4[3^8]$

Cell	Bond	cell	bond	vertex	edge
0.4695 ± 0.0858 -3.5574×10^{-5} 1.4884×10^{-4}	0.3044 ± 0.0429 2.8967×10^{-5} 6.8367×10^{-6}			0.7194 ± 0.0209 -5.8481×10^{-6} 6.5739×10^{-7}	0.6407 ± 0.0296 2.8990×10^{-6} 2.0123×10^{-6}
2[181], 4[265], 5.5691 5.6453 12[313] 5.6741	4[748], 12[888] 10.5294 10.5946			1[440], 2[624], 2.7273 2.7692 6[728] 2.7857	2[600], 2[864], 3.7333 3.7778 6[1014] 3.7949

$3_3[3^3]8_2[3^8]$

Cell	Bond	cell	bond	vertex	edge
0.4903 ± 0.1015 1.2205×10^{-4} 2.1262×10^{-4}	0.2926 ± 0.0281 -1.8541×10^{-5} 1.9682×10^{-6}			0.7447 ± 0.0329 2.1925×10^{-7} 2.8427×10^{-6}	0.7019 ± 0.0270 -6.4679×10^{-6} 8.9933×10^{-7}
2[166], 4[301], 5.1928 5.3953 12[476] 5.5168	4[812], 12[1313] 11.0788 11.2749			2[381], 2[668], 2.8346 2.8743 6[1035] 2.8986	2[540], 2[960], 3.8444 3.8833 6[1500] 3.9067

$3_3[3^3]9_3[3^9]_{\mathrm{III}}$

Cell	Bond	cell	bond	vertex	edge
0.4607 ± 0.0586 7.9409×10^{-5} 3.3840×10^{-5}	0.2483 ± 0.0258 4.8742×10^{-6} 7.4553×10^{-7}			0.7482 ± 0.0264 2.4849×10^{-6} 1.2161×10^{-6}	0.7049 ± 0.0195 -1.0461×10^{-5} 1.0354×10^{-6}
2[83], 4[357], 4.6988 5.3613 12[438] 5.4247	4[957], 12[1188] 12.0878 12.1785			2[212], 2[808], 2.6792 2.8317 6[980] 2.8469	2[284], 2[1144], 3.6620 3.8322 6[1395] 3.8480

$4_3[3^4]10_6[3^{10}]_{\mathrm{I}}$

Cell	Bond	cell	bond	vertex	edge
0.5286 ± 0.0511 5.4350×10^{-5} 1.7402×10^{-5}	0.2991 ± 0.0319 -8.0790×10^{-6} 2.7687×10^{-6}			0.7789 ± 0.0386 8.7871×10^{-6} 4.7653×10^{-6}	0.7173 ± 0.0391 -1.2114×10^{-4} 1.8197×10^{-5}
8[188], 4[295], 5.3085 5.4441 12[426] 5.5352	8[499], 4[803], 11.2826 11.5691 12[1179] 11.7574			4[429], 2[656], 2.6853 2.7439 6[931] 2.7841	4[576], 2[900], 3.8333 3.8667 6[1296] 3.8889

$3_3[3^3]9_3[3^9]_{\mathrm{I}}$

Cell	Bond	cell	bond	vertex	edge
0.5206 ± 0.0551 -9.0646×10^{-6} 1.7837×10^{-5}	0.2987 ± 0.0308 7.4580×10^{-6} 1.7780×10^{-6}			0.7532 ± 0.0296 -8.9673×10^{-6} 1.3152×10^{-6}	0.7063 ± 0.0222 -4.5643×10^{-6} 7.3255×10^{-7}
4[259], 4[473], 5.3977 5.5518 6[751] 5.6431	4[699], 4[1313], 11.9371 12.2224 6[2119] 12.3870			2[594], 2[1048], 2.8485 2.8855 6[1630] 2.9080	2[846], 2[1512], 3.8440 3.8836 6[2370] 3.9072

$4_2[3^4]10_4[3^{10}]$

Cell	Bond	cell	bond	vertex	edge
0.4726 ± 0.0684 1.0814×10^{-4} 8.0038×10^{-5}	0.2905 ± 0.0633 5.5163×10^{-4} 1.3835×10^{-4}			0.7341 ± 0.0316 -8.0603×10^{-6} 2.0508×10^{-6}	0.6957 ± 0.0303 7.5952×10^{-7} 1.6121×10^{-6}
4[309], 4[556], 5.3722 5.5288 12[875] 5.6229	4[830], 4[1537], 11.3181 11.6695 12[2460] 11.8756			2[670], 2[1181], 2.9015 2.9263 6[1836] 2.9412	2[972], 2[1728], 3.9156 3.9363 6[2700] 3.9489

$4_3[3^4]10_6[3^{10}]_{\mathrm{II}}$

Cell	Bond	cell	bond	vertex	edge
0.4958 ± 0.0796 1.3909×10^{-4} 8.0344×10^{-5}	0.3020 ± 0.0381 3.9193×10^{-5} 6.3920×10^{-6}			0.7490 ± 0.0461 1.0834×10^{-4} 1.4492×10^{-5}	0.6751 ± 0.0199 6.2752×10^{-7} 3.7299×10^{-7}
4[325], 4[325], 5.3292 5.3292 12[532] 5.4737	4[866], 4[866], 11.5727 11.5727 12[1456] 11.8187			2[748], 2[748], 2.8369 2.8369 6[1187] 2.8711	2[1061], 2[1061], 3.8624 3.8624 6[1704] 3.8920

$4_2[3^4]8_2[3^8]$

Cell	Bond	cell	bond	vertex	edge
0.5102 ± 0.0568 4.1767×10^{-5} 3.0853×10^{-5}	0.3183 ± 0.0405 -9.0691×10^{-6} 7.8890×10^{-6}			0.7202 ± 0.0493 -1.1640×10^{-4} 2.1475×10^{-5}	0.6953 ± 0.0307 1.5917×10^{-5} 1.8233×10^{-6}
4[259], 4[385], 5.2741 5.4026 12[535] 5.4916	4[683], 4[1040], 10.3309 10.5096 12[1469] 10.6331			2[613], 2[884], 2.8189 2.8507 6[1203] 2.8728	2[864], 2[1260], 3.8356 3.8651 6[1728] 3.8854

$4_3[3^4]8_3[3^8]_{\mathrm{I}}$

Cell	Bond	cell	bond	vertex	edge
0.4577 ± 0.0761 -2.5782×10^{-5} 8.4655×10^{-5}	0.2910 ± 0.0421 -1.6419×10^{-5} 7.8109×10^{-6}			0.6887 ± 0.0279 8.0593×10^{-7} 7.8808×10^{-7}	0.6532 ± 0.0340 -2.5506×10^{-5} 3.2110×10^{-6}
4[195], 4[356], 5.3231 5.4944 12[487] 5.5647	4[519], 4[978], 10.2736 10.5562 12[1355] 10.6657			2[454], 2[797], 2.8282 2.8708 6[1074] 2.8864	2[642], 2[1144], 3.8349 3.8759 6[1550] 3.8903

$4_3[3^4]8_3[3^8]_{II}$

Cell	Bond	cell	bond	vertex	edge
0.5030 ± 0.0603 2.6397×10^{-5} 4.0377×10^{-5}	0.3166 ± 0.0334 -1.3936×10^{-5} 2.7675×10^{-6}			0.7385 ± 0.0255 -2.6605×10^{-6} 9.6100×10^{-7}	0.6718 ± 0.0369 1.8952×10^{-6} 3.1533×10^{-6}
4[219], 4[419], 5.3151 5.4988 12[683] 5.6047	4[582], 4[1152], 10.2818 10.5799 12[1914] 10.7461			2[517], 2[947], 2.7505 2.8131 6[1505] 2.8505	2[711], 2[1332], 3.8312 3.8769 6[2145] 3.9030

$4_3[3^4]8_3[3^8]_{III}$

Cell	Bond	cell	bond	vertex	edge
0.4931 ± 0.0560 1.5620×10^{-4} 3.1433×10^{-5}	0.2927 ± 0.0320 3.0191×10^{-5} 6.4721×10^{-6}			0.7294 ± 0.0348 4.5084×10^{-6} 2.5226×10^{-6}	0.6730 ± 0.0289 -2.7067×10^{-5} 3.4615×10^{-6}
4[212], 4[410], 5.3491 5.5268 12[594] 5.6061	4[567], 4[1133], 10.3069 10.6019 12[1665] 10.7267			2[503], 2[927], 2.8151 2.8652 6[1317] 2.8853	2[708], 2[1328], 3.8107 3.8630 6[1900] 3.8832

$4_4[3^4]7_2[3^7]_{II}$

Cell	Bond	cell	bond	vertex	edge
0.4761 ± 0.0432 -7.3074×10^{-5} 9.5057×10^{-6}	0.3174 ± 0.0396 3.7498×10^{-5} 6.2114×10^{-6}			0.7144 ± 0.0228 7.2116×10^{-6} 6.1119×10^{-7}	0.6440 ± 0.0371 2.2191×10^{-5} 3.8797×10^{-6}
8[345], 12[486] 5.5710 5.6379	8[961], 12[1370] 10.0208 10.1241			4[767], 12[1063] 2.7901 2.8222	4[1070], 12[1500] 3.8579 3.8800

$3_3[3^3]12_6[3^{12}]$

Cell	Bond	cell	bond	vertex	edge
0.5077 ± 0.0896 4.5180×10^{-4} 3.2202×10^{-4}	0.2640 ± 0.0333 2.5513×10^{-6} 3.2552×10^{-6}			0.8273 ± 0.0209 1.3947×10^{-5} 1.6592×10^{-6}	0.7269 ± 0.0313 -2.2183×10^{-6} 1.6715×10^{-6}
4[154], 4[223], 4.5065 4.7444 4[374], 12[520] 5.0214 5.1654	4[347], 4[529], 13.9135 14.2949 4[939], 12[1343] 14.7178 14.9248			2[359], 2[507], 2.8078 2.8402 2[827], 6[1133] 2.8730 2.8914	2[504], 2[720], 3.8333 3.8611 2[1188], 6[1638] 3.8889 3.9048

$4_4[3^4]7_2[3^7]_{I}$

Cell	Bond	cell	bond	vertex	edge
0.4774 ± 0.0624 -1.6958×10^{-4} 4.8309×10^{-5}	0.3140 ± 0.0390 3.5112×10^{-5} 9.6223×10^{-6}			0.7305 ± 0.0263 7.4362×10^{-6} 1.0424×10^{-6}	0.6504 ± 0.0349 1.7310×10^{-5} 3.0145×10^{-6}
8[403], 12[554] 5.4591 5.5379	8[1100], 12[1534] 10.0236 10.1213			4[1050], 6[1421] 2.4686 2.4828	4[1296], 6[1764] 3.8704 3.8889

$5_4[3^5]7_4[3^7]_{I}$

Cell	Bond	cell	bond	vertex	edge
0.5112 ± 0.0412 -5.3542×10^{-5} 8.4833×10^{-6}	0.3406 ± 0.0207 2.1641×10^{-6} 5.0113×10^{-7}			0.7029 ± 0.0507 -5.7814×10^{-5} 1.4169×10^{-5}	0.6645 ± 0.0300 9.3355×10^{-6} 1.5205×10^{-6}
4[173], 4[450], 5.3179 5.5733 12[729] 5.6653	4[460], 4[1254], 9.4391 9.7895 12[2065] 9.9090			2[411], 2[1003], 2.7932 2.8654 6[1590] 2.8931	2[574], 2[1437], 3.8188 3.8859 6[2300] 3.9096

$3_1[4^3]5_1[4^5]_{I}$

Cell	Bond	cell	bond	vertex	edge
0.4003 ± 0.0534 -1.4626×10^{-5} 1.5617×10^{-5}	0.2368 ± 0.0198 -5.2284×10^{-6} 3.5502×10^{-7}	0.5786 ± 0.0400 3.9800×10^{-5} 5.2273×10^{-6}	0.4595 ± 0.0398 -2.9462×10^{-5} 5.1123×10^{-6}	0.6184 ± 0.0356 -1.2175×10^{-6} 2.6728×10^{-6}	0.5109 ± 0.0531 2.0112×10^{-5} 1.7919×10^{-5}
4[508] , 6[795] 7.37kk01 7.4943	4[1872], 6[2979] 14.1731 14.3411	4[508], 6[795] 3.7835 3.8264	4[961], 6[1521] 6.1748 6.2406	4[569], 6[871] 3.5993 3.6739	4[1024], 6[1600] 5.6738 5.7387

$3_1[4^3]5_1[4^5]_{II}$

Cell	Bond	cell	bond	vertex	edge
0.4065 ± 0.0646 -9.8055×10^{-5} 3.7012×10^{-5}	0.2309 ± 0.0241 2.2560×10^{-6} 6.6307×10^{-7}	0.5898 ± 0.0633 -1.8568×10^{-6} 3.3297×10^{-5}	0.4394 ± 0.0536 -1.3300×10^{-4} 3.6224×10^{-5}	0.6423 ± 0.0381 -5.1659×10^{-5} 7.7919×10^{-6}	0.5049 ± 0.0271 -9.8556×10^{-7} 8.8048×10^{-7}
4[512], 6[648] 7.6328 7.6728	4[1954], 6[2486] 14.2027 14.2912	4[512], 6[648] 3.8750 3.8889	4[992], 6[1260] 6.2016 6.2349	4[545], 6[685] 3.7578 3.7839	4[1024], 6[1296] 5.8145 5.8349

$5_3[3^5]8_6[3^8]_{II}$

Cell	Bond	cell	bond	vertex	edge
0.4631 ± 0.0471 2.8969×10^{-5} 1.3943×10^{-5}	0.3164 ± 0.0255 3.0266×10^{-6} 1.2108×10^{-6}			0.6800 ± 0.0384 -9.9185×10^{-6} 4.4523×10^{-6}	0.6587 ± 0.0246 1.5506×10^{-6} 6.3039×10^{-7}
4[260], 4[675], 5.4308 5.6474 12[1095] 5.7205	4[706], 4[1906], 9.8669 10.1815 12[3132] 10.2854			2[608], 2[1490], 2.8191 2.8846 6[2368] 2.9071	2[857], 2[2149], 3.8273 3.8920 6[3442] 3.9140

$5_3[3^5]8_6[3^8]_{III}$

Cell	Bond	cell	bond	vertex	edge
0.5052 ± 0.0474	0.3211 ± 0.0274			0.7145 ± 0.0299	0.6469 ± 0.0242
-1.6178×10^{-6}	-9.4980×10^{-6}			8.1493×10^{-6}	-1.0841×10^{-5}
1.0710×10^{-5}	2.4172×10^{-6}			1.7096×10^{-6}	7.2095×10^{-7}
8[351], 12[814]	8[971], 12[2316]			4[804], 6[1782]	4[1144], 6[2580]
5.5328 5.6904	10.0062 10.2366			2.8458 2.8956	3.8531 3.9023

$5_2[3^5]12_{12}[3^{12}]$

Cell	Bond	cell	bond	vertex	edge
0.4897 ± 0.0718	0.3037 ± 0.0257			0.7384 ± 0.0258	0.6694 ± 0.0297
-1.4823×10^{-4}	-1.3145×10^{-5}			3.3765×10^{-6}	1.5898×10^{-5}
1.0656×10^{-4}	1.4342×10^{-6}			8.4513×10^{-7}	2.4608×10^{-6}
4[192], 8[530], 12[803]	4[498], 8[1457], 12[2244]			2[442], 4[1156], 6[1724]	2[627], 4[1675], 6[2514]
5.1875 5.4981 5.5890	10.8072 11.2972 11.4332			2.8371 2.8979 2.9165	3.8405 3.9021 3.9204

$5_4[3^5]7_4[3^7]_{II}$

Cell	Bond	cell	bond	vertex	edge
0.5001 ± 0.0342	0.3327 ± 0.0329			0.6973 ± 0.0235	0.6688 ± 0.0303
1.7585×10^{-5}	4.4678×10^{-5}			-7.2307×10^{-6}	2.4051×10^{-5}
3.4856×10^{-6}	5.2573×10^{-6}			5.9288×10^{-7}	2.0921×10^{-6}
8[371], 12[541]	8[1027], 12[1519]			4[835], 6[1194]	4[1195], 6[1722]
5.5364 5.6155	9.7254 9.8328			2.8623 2.8844	3.8661 3.8885

$5_3[3^5]8_6[3^8]_{I}$

Cell	Bond	cell	bond	vertex	edge
0.4867 ± 0.0459	0.3265 ± 0.0237			0.7108 ± 0.0200	0.6720 ± 0.0294
-1.0891×10^{-4}	-3.7362×10^{-6}			2.8929×10^{-6}	6.2044×10^{-6}
3.3218×10^{-5}	7.9808×10^{-7}			4.2792×10^{-7}	3.1424×10^{-6}
6[214], 8[535], 12[790]	4[577], 8[1502], 12[2245]			2[506], 4[1192], 6[1728]	2[731], 4[1762], 6[2573]
5.3925 5.6150 5.6835	9.8024 10.1278 10.2254			2.8261 2.8859 2.9051	3.9398 4.0204 4.0459

$4_2[3^4]12_6[3^{12}]$

Cell	Bond	cell	bond	vertex	edge
0.5117 ± 0.0960	0.2637 ± 0.0295			0.7861 ± 0.0215	0.7273 ± 0.0227
1.0580×10^{-4}	-4.6675×10^{-6}			7.1685×10^{-6}	1.1750×10^{-6}
2.1578×10^{-4}	1.9508×10^{-6}			4.3496×10^{-7}	7.1142×10^{-7}
4[200], 8[444], 12[629]	4[461], 8[1121], 12[1633]			2[474], 4[997], 6[1388]	2[664], 4[1428], 6[2002]
4.6100 5.0495 5.1924	11.1714 12.1802 12.4850			2.8017 2.8646 2.8847	3.8614 3.9048 3.9191

$4_2[3^4]18_{12}[3^{18}]$

Cell	Bond	cell	bond	vertex	edge
0.5249 ± 0.1168	0.2493 ± 0.0632			0.8497 ± 0.0234	0.7743 ± 0.0483
1.1603×10^{-3}	7.1820×10^{-4}			5.2173×10^{-6}	8.6384×10^{-5}
6.1479×10^{-4}	2.0594×10^{-4}			6.1183×10^{-7}	1.2578×10^{-5}
8[237], 4[428], 12[675]	8[576], 4[1101], 12[1794]			4[529], 2[929], 6[1441]	4[774], 2[1376], 6[2150]
4.8608 5.1449 5.3156	15.0868 15.8656 16.3155			2.8696 2.9020 2.9216	3.9664 4.0087 4.0344

$5_3[3^5]7_3[3^7]_{I}$

Cell	Bond	cell	bond	vertex	edge
0.4882 ± 0.0494	0.3335 ± 0.0310			0.7149 ± 0.0273	0.6495 ± 0.0278
-4.4634×10^{-5}	-1.2425×10^{-5}			1.6306×10^{-6}	5.9525×10^{-6}
1.5462×10^{-5}	2.4243×10^{-6}			9.8464×10^{-7}	1.4733×10^{-6}
4[170], 4[399]	4[455], 4[1112]			2[413], 2[907]	2[574], 2[1293]
5.3529 5.5739	9.4110 9.7392			2.7797 2.8512	3.8084 3.8716

$5_3[3^5]7_3[3^7]_{II}$

Cell	Bond	cell	bond	vertex	edge
0.4345 ± 0.0489	0.2930 ± 0.0243	0.4785 ± 0.0354	0.3239 ± 0.0187	0.6900 ± 0.0283	0.6421 ± 0.0289
8.3482×10^{-5}	-9.3058×10^{-6}	4.7566×10^{-5}	-4.1986×10^{-6}	-7.8267×10^{-6}	3.4481×10^{-6}
4.0009×10^{-5}	1.1454×10^{-6}	7.5326×10^{-6}	3.8717×10^{-7}	1.5860×10^{-6}	1.8973×10^{-6}
2[167], 2[61], 6[535]	2[161] 6[1670]	2[167], 2[61], 6[535]	2[480], 2[159], 6[1638]	2[386], 2[161], 6[1124]	2[535], 2[213], 6[1611]
5.8443 5.2787 6.2430	9.8137 11.3257	5.7485 5.2131 6.1234	10.4458 9.5849 11.0000	2.7720 2.6460 2.8665	3.8019 3.6901 3.8845

$3_2[4^3]5_2[4^5]_{\mathrm{I}}$

Cell	Bond	cell	bond	vertex	edge
0.4099 ± 0.0264	0.2271 ± 0.0154	0.5687 ± 0.0469	0.5028 ± 0.0283	0.6228 ± 0.0346	0.5022 ± 0.0343
4.7650×10^{-6}	2.1524×10^{-6}	2.6238×10^{-5}	1.1978×10^{-5}	-1.1730×10^{-5}	3.4916×10^{-5}
1.0783×10^{-6}	1.2777×10^{-7}	8.5056×10^{-6}	1.1260×10^{-6}	5.5785×10^{-6}	3.2050×10^{-6}
2[277], 2[379], 7.3285 7.4248 6[497] 7.4970	2[1407], 6[1863] 14.0640 14.1857	2[277], 2[379], 3.7473 3.7836 6[497] 3.8109	2[519], 2[717], 6.0308 6.1004 6[947] 6.1521	2[325], 2[435], 3.5446 3.6046 6[561] 3.6506	2[576], 2[784], 5.6285 5.6811 6[1024] 5.7207

$3_2[4^3]5_2[4^5]_{\mathrm{II}}$

Cell	Bond	cell	bond	vertex	edge
0.4322 ± 0.0512	0.2344 ± 0.0171	0.5638 ± 0.0590	0.4749 ± 0.0440	0.6260 ± 0.0368	0.5468 ± 0.0270
5.8373×10^{-5}	-1.5204×10^{-6}	4.8226×10^{-6}	-5.1510×10^{-5}	-2.9282×10^{-5}	8.1058×10^{-6}
1.9444×10^{-5}	2.2157×10^{-7}	1.5253×10^{-5}	1.0399×10^{-5}	6.7165×10^{-6}	1.6103×10^{-6}
2[276], 2[496], 7.1812 7.3831 6[780] 7.5051	2[1831], 6[2927] 14.1038 14.2890	2[276], 2[496], 3.6739 3.7540 6[780] 3.8026	2[507], 2[931], 5.9961 6.1267 6[1483] 6.2036	2[313], 2[545], 3.6422 3.7284 6[841] 3.7812	2[570], 2[1016], 5.6702 5.7520 6[1590] 5.8013

$3_3[4^3]5_3[4^5]$

Cell	Bond	cell	bond	vertex	edge
0.5029 ± 0.0759	0.2504 ± 0.0233	0.6238 ± 0.0601	0.5029 ± 0.0359	0.6645 ± 0.0414	0.5801 ± 0.0306
8.6335×10^{-6}	-3.9666×10^{-6}	-1.3226×10^{-4}	1.1319×10^{-4}	1.1038×10^{-4}	-3.3375×10^{-5}
1.4737×10^{-4}	8.6134×10^{-7}	3.8526×10^{-5}	1.9717×10^{-5}	1.6270×10^{-5}	3.6510×10^{-6}
4[161], 2[315], 6.9565 7.2317 6[473] 7.3700	2[1139], 6[1743] 13.8191 14.0425	6[161], 2[315], 3.5404 3.6698 6[473] 3.7294	4[285], 6[578], 5.8175 6.0104 6[882] 6.1020	4[200], 2[369], 3.3600 3.5122 6[539] 3.5918	4[336], 2[648], 5.4881 5.6204 6[968] 5.6880

$3_1[4^3]6_2[4^6]_{\mathrm{I}}$

Cell	Bond	cell	bond	vertex	edge
0.4339 ± 0.0259	0.2335 ± 0.0200	0.5468 ± 0.0321	0.4235 ± 0.0152	0.6418 ± 0.0638	0.5715 ± 0.0431
4.6901×10^{-7}	-3.2315×10^{-6}	3.4990×10^{-5}	2.5190×10^{-6}	-1.1180×10^{-4}	4.9165×10^{-5}
6.9917×10^{-7}	3.6617×10^{-6}	4.5289×10^{-6}	1.5119×10^{-7}	4.6849×10^{-5}	7.7330×10^{-6}
2[760], 6[1190] 7.5263 7.6202	2[2860], 6[4534] 15.2503 15.4027	2[760], 6[1190] 3.8342 3.8672	2[1457], 6[2301] 6.7138 6.7718	2[833], 6[1281] 3.6879 3.7471	2[1536], 6[2400] 5.7513 5.8008

$3_1[4^3]6_2[4^6]_{\mathrm{II}}$

Cell	Bond	cell	bond	vertex	edge
0.4740 ± 0.0681	0.2803 ± 0.0276	0.6333 ± 0.0534	0.4785 ± 0.0260	0.6961 ± 0.0552	0.6132 ± 0.0465
-1.5376×10^{-5}	-1.1319×10^{-5}	-1.4476×10^{-4}	1.8163×10^{-5}	2.5890×10^{-5}	6.0868×10^{-5}
4.0570×10^{-5}	1.3703×10^{-6}	2.7149×10^{-5}	1.3662×10^{-6}	2.2393×10^{-5}	1.1922×10^{-5}
2[245], 2[477], 7.0531 7.3040 6[715] 7.4294	2[1742], 6[2656] 14.8576 15.0715	2[245], 2[477], 3.6408 3.7400 6[715] 3.7874	2[446], 2[892], 6.3184 6.5135 6[1354] 6.6041	2[284], 2[532], 3.5493 3.6541 6[782] 3.7136	2[504], 2[972], 5.6349 5.7243 6[1452] 5.7741

$3_2[4^3]6_4[4^6]$

Cell	Bond	cell	bond	vertex	edge
0.4540 ± 0.0687	0.2265 ± 0.0369	0.6160 ± 0.0685	0.4908 ± 0.0346	0.6320 ± 0.0532	0.5527 ± 0.0434
7.8079×10^{-5}	-1.5535×10^{-5}	-1.8755×10^{-4}	-1.2315×10^{-5}	1.0557×10^{-4}	1.0264×10^{-5}
4.6624×10^{-5}	2.8752×10^{-6}	5.5163×10^{-5}	2.5752×10^{-6}	2.0627×10^{-5}	1.0088×10^{-5}
2[203], 2[440], 6.9754 7.2955 6[550] 7.3673	2[1605], 6[2026] 14.9533 15.0642	2[203], 2[440], 3.6059 3.7318 6[550] 3.7600	2[366], 2[821], 6.4645 6.6358 6[1034] 6.6731	2[238], 2[491], 3.5294 3.6660 6[607] 3.6969	2[420], 2[900], 5.8095 5.9689 6[1122] 6.0036

$3_3[4^3]6_6[4^6]$

Cell	Bond	cell	bond	vertex	edge
0.4042 ± 0.0580	0.2281 ± 0.0277	0.5958 ± 0.0567	0.4698 ± 0.0313	0.6353 ± 0.0449	0.5210 ± 0.0370
3.6150×10^{-5}	2.6810×10^{-6}	-4.2796×10^{-5}	3.1642×10^{-5}	-2.0589×10^{-5}	7.9238×10^{-6}
1.8890×10^{-5}	7.5233×10^{-7}	1.4656×10^{-5}	4.6775×10^{-6}	8.7591×10^{-6}	2.4104×10^{-6}
2[230], 2[490], 6.9391 7.2735 6[656] 7.3720	2[1782], 6[2418] 14.8911 15.0488	2[230], 2[490], 3.5304 3.6816 6[656] 3.7256	2[406], 2[902], 6.5714 6.7095 6[1222] 6.7496	2[267], 2[543], 3.5955 3.7127 6[717] 3.7490	2[480], 2[1008], 5.6417 5.7500 6[1344] 5.7827

$3_1[4^3]8_4[4^8]$

Cell	Bond	cell	bond	vertex	edge
0.4525 ± 0.0780	0.2122 ± 0.0230	0.5237 ± 0.0627	0.3807 ± 0.0401	0.7010 ± 0.0470	0.6086 ± 0.0679
7.1913×10^{-4}	-9.6553×10^{-6}	-1.0143×10^{-5}	3.5308×10^{-6}	-7.5034×10^{-5}	1.6014×10^{-4}
1.4678×10^{-4}	8.2295×10^{-7}	3.2798×10^{-5}	5.4378×10^{-6}	1.2495×10^{-5}	3.8208×10^{-5}
2[238], 2[490], 7.0336 7.3224 6[708] 7.4350	2[1794], 6[2632] 16.9197 17.1071	2[238], 2[490], 3.6555 3.7592 6[708] 3.7994	2[435], 2[921], 7.3931 7.5831 6[1345] 7.6550	2[288], 2[561], 3.4028 3.5651 6[793] 3.6318	2[490], 2[1000], 5.4898 5.6420 6[1440] 5.7014

$3_1[5^3]4_2[5^4]_{\mathrm{I}}$

Cell	Bond	cell	bond	vertex	edge
0.4047 ± 0.0637 6.1235×10^{-5} 3.1306×10^{-5}	0.2185 ± 0.0212 5.5289×10^{-6} 5.4838×10^{-7}	0.6180 ± 0.0491 6.5660×10^{-5} 1.1080×10^{-5}	0.4930 ± 0.0327 1.4895×10^{-5} 2.1733×10^{-6}	0.5516 ± 0.0469 4.3305×10^{-5} 1.3841×10^{-5}	0.4101 ± 0.0301 1.2816×10^{-5} 2.1196×10^{-6}
2[252], 2[570], 8 8.3298 6[777] 8.4247	2[2374], 6[3273] 16.0126 16.1577	2[252], 2[570], 3.7143 3.8105 6[777] 3.8378	2[468], 2[1086], 6.3932 6.5157 6[1491] 6.5500	2[281], 2[613], 4.4982 4.6591 6[827] 4.7062	2[632], 2[1428], 7.4525 7.6345 6[1946] 7.6865

$3_1[5^3]4_2[5^4]$

Cell	Bond	cell	bond	vertex	edge
0.3457 ± 0.0484 8.5340×10^{-5} 2.5074×10^{-5}	0.1820 ± 0.0200 -5.3816×10^{-6} 6.0154×10^{-7}	0.6226 ± 0.0443 -5.2469×10^{-5} 1.2122×10^{-5}	0.5695 ± 0.0477 -1.2139×10^{-4} 1.8902×10^{-5}	0.5527 ± 0.0393 -2.9005×10^{-5} 1.0732×10^{-5}	0.4162 ± 0.0419 -3.2254×10^{-5} 7.5538×10^{-6}
8[213], 2[380], 8.9202 9.1842 6[595] 9.3445	6[950], 2[1745], 16.8147 17.2218 6[2780] 17.4626	8[213], 2[380], 3.1080 3.1632 6[595] 3.1966	8[331], 2[601], 4.4411 4.5324 6[951] 4.5868	8[169], 2[289], 4.2604 4.4291 6[441] 4.5351	8[360], 2[640], 7.3444 7.5062 6[1000] 7.6040

$3_2[5^3]4_4[5^4]$

Cell	Bond	cell	bond	vertex	edge
0.3672 ± 0.0405 -6.6453×10^{-5} 1.1174×10^{-5}	0.1766 ± 0.0243 -3.2608×10^{-6} 1.0035×10^{-6}	0.6265 ± 0.0398 7.1192×10^{-5} 7.0957×10^{-6}	0.5822 ± 0.0505 2.3922×10^{-5} 2.8516×10^{-5}	0.5413 ± 0.0515 -9.8715×10^{-5} 1.9319×10^{-5}	0.4101 ± 0.0536 9.3126×10^{-5} 3.1517×10^{-5}
4[168], 4[270], 8.7143 8.9778 6[396] 9.1515	2[732], 2[1212], 16.5929 16.9868 6[1812] 17.2395	4[168], 2[270], 3.0357 3.0963 6[396] 3.1364	4[255], 2[418], 4.4078 4.4928 6[621] 4.5475	4[141], 4[216], 4.3688 4.4907 6[307] 4.5733	4[308], 4[485], 7.2662 7.4144 6[702] 7.5128

Table 4.7 *Percolation statistics of the 2-homohedral tilings. Each item of the statistics are, from top to bottom, mean ± standard deviation, second moment, and third moment. Similarly each of the items in each box contains the number of runs [the number of network components] and the coordination number.*

The advantage of sparse- over dense networks described by Burt (1992) is an idea similar to that of the benefit of decentralisation and local autonomy in politics. According to him, size still matters but the cost of maintaining the size is also important. The idea can be very general; dense networks are virtually worthless monitoring devices, while sparse networks give more information benefits. Taking opportunity costs into account, the rate of return of a dense network is lower than that of a sparse network. The idea can be applied to a strategic network expansion which is crucial for sales persons and jobs hunting alike.

Something similar to this comes up in a variety of fields. In Sociology it is sometimes called the *strength of weak ties*. A tie between two things is weaker the less each of them has in common with the other. It is well known, *i.e.* the origin unknown, that best friends are usually those who have least in common with each other. It would be interesting to trace this idea back to where it first appeared in literature.

The study above shows that the percolation probability is not a function of the coordination number of the network alone. I suspect that it is a function of both this and the coordination number of the dual lattice of the network, that is $p_c = \mathrm{f}(x_c, x_v)$. In other words, p_c may be a function of not only the connectivity but also the tortuosity of the network.

§ 4.10 Cosmology

The universe is made up of superclusters of galaxies, each of which contains local clusters. Each local cluster contains galaxies, each of which contains stars. When stars explode, gas bubbles are formed which expand to meet one another, forming walls of materials in the manner of the Voronoi tessellation in three dimensions. Within the planets around each stars there exist yet other structures which can be similarly represented by the Voronoi tessellation.

Superclusters are like endless cobwebs covering the space. The Andromeda galaxy, the Sculptor group, the Virgo cluster, and the M81 group are examples of the members of the Local Supercluster in which our Milky Way galaxy reside. The Pisces-Perseus supercluster is 250 million, and the Hercules supercluster 500 million light-years away from us.

Assuming the big bang origin of the universe, the defects originated in the phase transition at the Big Bang could play an important role in the formation of galaxies and their clusters and superclusters. Tom Kibble predicted this in 1976 (*cf* Croswell, 1995), and gave three types of possible defects, namely the point-like monopoles, the line-like cosmic strings, and the plane-like domain walls.

Van de Weygaert (1989) studied the statistics of 3-d Voronoi tessellations for the purpose of understanding the structure of the universe. Here *neighbours* are defined by a common polygonal face whereas *full neighbours* have an additional requirement that the line joining nuclei intersects this common face. Taking a linear section through d-d, $d \geq 2$, the mean length is

$$\langle \lambda \rangle = \frac{d\Gamma\left(d-\frac{1}{2}\right)\Gamma\left(\frac{d+1}{2}\right)^2}{(d-1)!\rho^{\frac{1}{d}}2\Gamma\left(\frac{d}{2}+1\right)^{2-\left(\frac{1}{d}\right)}\Gamma\left(2-\frac{1}{d}\right)}, \qquad (\mathbf{13})_{\text{iv}}$$

where ρ is the number density, in other words nucleus density.

In three dimensions we need at least two points in order to establish the distance of an object by a geometrical method. Because the cosmic scale is so large compared with the scale of the solar system and Earth, all the observation points we may choose become the same and only one point in practice. Therefore the determination of the distance to extragalactic objects, which is crucial in cosmology, can only be achieved by the various means of observations from a single point in space. This is a limitation that has caused the establishment of distance to be a hot, long-standing controversial issue. As the universe is expanding with the receding velocity of objects increases with their distance away from us, and as this velocity can be accurately determined by the amount of redshift in the light of these objects observed, the study of distance becomes the study of a single, universal constant called the Hubble constant. This is only a constant in theory not in practice. All numerical values of the Hubble constant used in literature are fictitious to some degree. If its exact value is known, then the accurate distance to any cosmic object is simply its radial velocity, *i.e.* its velocity away from us, divided by this Hubble constant. The reciprocal of the Hubble constant is called the Hubble time. The importance of the Hubble constant, together with the difficulty in finding it, make the observational and the theoretical parts of cosmology inseparable from each other. They are also the cause of the proliferation of the modern literature on Astrophysics. One example of studies of the extragalactic distance scale is that reported by de Vancouleurs (1993) which includes among the objects studied the following objects belonging to the Local Group, namely the Large Magellanic Cloud, M31, M33, NGC 3109, as well as the objects beyond the Local Group, namely NGC300, IC4182, NGC2403, NGC5128, M81, M101, M104 (NGC4594), and NGC4571.

§ 4.11 Philosophy

Philosophy precedes science, and so the case is with Voronoi tessellation by more than two hundred years. In his philosophical scientific work, Descartes (1644) imagines the universe as a tessellation of vortices surrounding stars. According to the pictures he gave, a few of which are duplicated many times over, the positions of these stars are random. Among these, the most abundant picture resembles a two dimensional picture of biological cells or a Voronoi graph. His idea is that stars always position themselves as far apart from one another as possible. At any instance each of these bodies resides within a partition of its own, where it acts as the centre of a vortex around which everything inside the partition revolves.

However, when one would like to track the movement made by a body one may consider the rest as having a fixed position. Starting from anywhere within a partition, the body under consideration moves towards a boundary that lies across its trajectory. Once the boundary is reached, it continues moving along that boundary until it reaches a vertex from the requirement of keeping itself furthest from surrounding stars, and then it chooses the one of the two choices of boundary branches which would make it deviate the least from its present direction. In this way a body snakes its way further and further from its starting position. Although the pictures that Descartes gave were in two dimensions, there is no doubt that he means, and was talking about a three dimensional space. Therefore these two dimensional pictures of his can either be a 2–d section of a three dimensional space, or simply a 2–d representation of it.

As is the case with the study of incompressible fluid flow by visualising it as a displacement of small rigid spherical particles, whose treatment within the same treatise of his predates the ground-breaking theoretical works of Hele-Shaw (1899) by more than two and a half centuries, the idea he has given about the movement of heavenly bodies predates the idea of van de Weygaert where the similarity is obvious in both the treatment of, as well as the objects considered. Van de Weygaert and Icke (1989) considers the movements of galaxies within voids with sizes several orders larger than themselves. In their simulation, each galaxy moves outwards from anywhere within a void until it reaches a wall where, because of the interaction with other galaxies coming from a neighbouring void which it meets, it loses its radial velocity and continues to move with reduced speed along the plane of the wall. Eventually it will come to an edge where it loses a second part of its velocity components, again thanks to other galaxies this time coming from the other two of the three faces which comes together at the edge. The last component of velocity it has left then takes it as far as the first vertex it should meet and no further. Having lost all the remaining energy to travel further, again through the encounter with other galaxies this time from three other edges which meet with the edge along which it has been travelling, it joins other galaxies to form a cluster which keeps growing all the time.

If the idea of Descartes is that of stars travelling along a Voronoi graph, then the idea of van de Weygaert and Icke represents a universe which is lumpy at Voronoi vertices. Seemingly different, both ideas are very similar. Both consider partitions containing nuclei. The only difference seems to be that in the former case these nuclei are massive bodies, while in the latter they are voids. Thus in a way the two can be said to be dual to each other. Furthermore, the scale being considered seems to be by many orders different. Descartes' vortex idea can easily apply to the scale size of a galaxy, while the vertex idea proposed by van de Weygaert and Icke seems to agree well with observations made in scale unimaginably large compared with that of the other, since they obviously consider giant clusters of over a thousand galaxies in size as being vertices, in other word points.

Voronoi tessellation thus came up philosophically first and then science. Percolation theory, on the other hand, is itself philosophical. Not only that it often sheds light on the underlying reason of many things, but it sometimes is the core philosophy in its own right.

Karma is the central philosophy of Buddhism. In East Asia there are similar sayings which mean *Karma is deed* (*cf Lan. Kamṁ ṗẹn kạrkdaṃ* and *Th. Kamr gụe kạrkradaṃ*). The word *karma* in English adopts a more practical one among the plethora of implications or nuances in meaning, which says that it is the sum of actions of a person in one of his successive lives, which decides his fate in the next. Another meaning puts it simply as destiny or fate. Karma and enlightenment are the two things around which Buddhism philosophy revolves. Karma entails enlightenment. In other words, the latter is a subset of the former. To put it plainly, enlightenment is Pcwhile karma is the corresponding Pc-process.

To me the teaching of Buddha is about percolation, and hence the idea was discovered by Buddha whom I consider a great philosopher, at least more than two millenia and five centuries before it was rediscovered in Physics of our time. Now it may seem ironic after having said that the

Buddhist philosophy is percolation if I as professedly a christian should say that I have converted myself to the religion as a result of a percolation, having been among buddhists since I was born and having led my life as one for twenty years. When I left a job with a good prospect behind me to come to England in 1994 I said to myself that having been forsaken by two of the women I most love within a period of four years it did not seem that life was no longer worth living and making money seemed all very well except that I had no one now to make it for. Then in the sombre atmosphere of Manchester I read the two paperback translated volumes of *Les Misérables* by Victor Hugo which suddenly turned me into a christian. I think of this experience now as the percolative workings of various elements. Having never been the best student in my class, except sometimes in the subject of the English language, I had found myself often in ungraceful academic situations from 1984 to 1991 when I received my first degree. My study had given me much stress and the life in Bangkok with its best percolated traffic jam in the world was no help. Due to some historical reasons and the difference in the languages a lanna in Bangkok is usually considered as belonging to a second class population. This combined with the heart having been broken from the cause mentioned and from the background of having lived with a Kiwi western family and the fact that I have always studied in a christian school which had provided me with the necessary background and belief, it was not difficult for me to profess myself a christian, but Hugo still has the final credit for it. I spent close to twelve years studying at the Montfort College in Chiangmai, Thailand. The college was named after Saint Louis Marie Grignion de Montfort (1673–1716) who founded the Congregation of the Brothers of Christian Institute of St. Gabriel in 1705 to which my school belongs.

I still believe in the philosophy of Buddha. I see Buddhism as a philosophy the idea of which is an analogy to percolation, but I do not see it as a religion, which to me necessarily must start from a myth and have a god. Buddha, however, is no god, and taught his findings which were results of the Enlightenment he experienced as a philosophy from the start. What myths there are in Buddhism now have invariably been added hundreds if not thousands of years later in accord with the tastes and methods of the rulers and preachers. I do not consider this kind of myths as real, and consider them superficial and unnecessarily for as good a philosophy as Buddhism.

In doing a PhD I find that I have to make war with my own ignorance and unproductiveness. But no doubt at each step forward and as each new idea comes in I am a step closer to my goal. And if I think that I should mind more about the quality of my work than about the end result, then I am using the same argument that I use in § 8.2 to explain the economic potential of a country. The internet and plastic money or credit cards may be factors which help the birth of the Euro possible. But the potential and connectivity has already gathered up considerably since the fall of the Iron Curtain and the reunification of Germanywith its amazing one to one exchange from the eastern to the western Deutsch Marks. As much as that the fall of both the Iron Curtain and the Berlin Wall may owe much to the internet and the spread of information by the media the internal flaws within the communist regimes of the Eastern Block countries must have also played a great part. Because otherwise why the same has not happened to the West saturated with the media? The West with its distributed government, a system inherited from America, has an advantage of more robust a system. This is a strong point of democracy. Socialism also has its unique strong point regarding the feeling of security in individuals which helps making the system robust. So as the flaws inside a percolating cluster collectively break it down, another structure is formed up from the ash like a phoenix. As much as individual parts form a single structure, *like brooks make river, rivers run to sea*, so do impurities within a structure break it up the same way as flaws do the percolating cluster. The spread of terrorism from the middle east owes much to the connectivity of air routes.

If we accept Plato's (360 BC)argument that men are the same because all human souls are the same, then any big difference in the cultures of the world can only be satisfactorily explained by using the percolation theory. If all human souls are the same and souls are more important than bodies, then all men are the same and all societies being comprised of equivalent people will necessarily have an identical structure or structural basis. Intrinsic attributes of a structure, for example the culture of a country or the way of life of its population, can take on various forms according to the phase which it has arrived at through the percolation under the various factors which includes historical and geographical influences. One example is the stark contrast on the value given to winning and losing between the western and the eastern cultures. As I see it, for the former the emphasis is on winners, which is reflected in sayings like *winner takes all*, as contrasted with the emphasis in the latter on losers, for example the thai saying *win and become the devil, lose and become a saint* [bāe pen bra, jàna pen mạr]. As another example, let us consider the japanese saying *akirameta yo nanisama akirameta, akiramerarenu to akirameta*, which are seemingly easy but

which has eluded some distinguished western experts in the Japnese language and culture. Months after having resigned from a PhD course in Japan, Keiko Saitōsent me this poem in an email. As a keen archer in kyūdō, that ancient japanese art which uses bows with lengths of 212 to 245 cm long, I think I can grasp the meaning. But not long afterwards, while browsing through the books in a japanese library in Bangkok I came across another translated version of the poem which is completely different to what I think it means. In his translation, Blyth (1959) has written, "I have resigned myself! Resigned yourself to what? I have resigned myself to never being resigned." The reason I say it is different is that my own version is the following soliloquy, "You are beaten! In what way do you say I am beaten? If you said that you shall never be defeated then in fact you have already lost." Kyūdō teaches one that results are nothing. This is not only true for kyūdou., but to all martial arts as well, provided that they are genuine. The training in any art of fighting is never to learn how to fight, but to train the subconscious mind such that it will be able to act by it self at the right time. The subconscious mind is similar to what Freud calls *superego*. In line with Plato's argument that the human souls are never evil, to martial arts this subconscious mind invariably carries the same attributes to that of the superego. In such perspective the Freudian id has no part to play and can be considered as being neither a subconscious mind nor on the same level with the superego; whenever pushed one never push back, and every fight is a self-protective one. The same is the case with kyūdō where there is a difference between an arrow hitting the target and the pride of a successful hit. An arrow hits the target best when it was already there before you took aim. Similar to the idea of complex superposition in the quantum mechanics theory, the free subsconscious mind superimposes and becomes one with the surrounding which includes the target as well as the path towards it through the air. The arrow thus runs home on its own accord. No one could have put it there; certainly not yourself, so there never is anything to be proud of. Only a few months earlier it was that I learnt the art of Kyūdō for the first time when on 14^{th} September 1997 in the 36^{th} Meguro-ku Sport Festival of the Meguro district in Tokyo I led our underdog B Team of three to win second. The other two members of the team were Arakawa Hiroshi and Suzuki Jirō. Looking back I can still remember how when one third through our round and having not scored a single point I suddenly feel one with the surrounding, totally oblivion to the sound of the audience behind us, and let go of the target with the result of which not only that I could break the egg but also the way the manner in which the arrow hit the target was perfect. I am proud to say now that eventhough that hit was the first and only one by me in that event, the other two of my teammates from that point onwards never again failed a single shot through to the end of our round. I wonder if it is not the case that, having been trained to be at ease and relaxed with the bow, my subconscious being, that thing inside me which still do not know much about, kicked in at such the right moment that percolated or communicated itself to my teammates, and then, being sure of having accomplished the task, left the scene. I am rather glad to be left anonymous in this way. To account for my claim of not having consciously aimed that transitional shot, for fifteen years none of my glasses has been adequately prescribed to let me see the bullseye clearly from that distance. Probably nobody will ever know that it was Kit Tiyapan whose name stands first among the three members of the B Team because, reluctant to have my name written in that outlandish katagana appear on three certificates for fear that it disgraces my teammates, I went by the name Zhāng Míng Lóng , what was given to me by my late grandfather and its sole user.

Waking up to the radio on 30^{th} May 2002 morning I heard on the BBC4 one writer as saying that international conflicts and crises occur because matters take on *momentum* which is sometimes difficult to reverse, that it can not be left alone to resolve like playing the russian roulette. I think the only thing that can solve an international, or internal, crisis is a counter momentum built up in the opposite direction. Some says that force stops dangerous mechanisms and that politics never work. No doubt any politics that has no strategies by which to reach at such counter momentum as needed will probably never work.

Life is a dream, or dreams are the other phases of life. If we consider dreams to be in the domains where our minds are in different phases from our awaking selves, then the analysis of the dreams would amount to the study of these various phases. Qualitative sayings like this are sometimes mistakenly written off as nonsensical in the field of science. What is meant when Sacks (1973), for instance, speaks of 'romantic science' being badly needed is that eventhough art can stand apart from science, science can not be separated from art. I do not think such 'romantic science' is that badly needed, at least not in the field of geometry, mathematics, or percolation. But then again, the first two of these are to me essentially the languages, while the last one the music,

and therefore also a kind of language, of nature. Music is the language of the mind and the thing that holds it together. Arts and music heal the mind. But they may not work, which makes them different from medicines. Something mysterious has to trigger before they do. As the *ear* of science they are the things which hold together the *eye* or structure of science.

Opposing forces in nature is an immemorial theme which recurs time and again. This dualism or the law of polarity shows itself, for example, in the eastern philosophy of the opposition between Yin and Yang. This theme came up again in the mid 20^{th} century in the name of percolation, the interplay between two opposing phases. The dynamic triad where the opposing *thesis* and *antithesis* come together and resolve each other into the *synthesis* is analogous with, for instance, the opposition between congested and free-flowing roads in the traffic network which resolve each other in the case where $p_c < 0.5$ into the band of synthesis having the width of $2(p_c - 0.5)$ and centred at 0.5 (§ 7.1).

The conceptualising power of man culminates in the recurrent themes found more prominently in creative minds (Hankins, 1997). These recurring themes represent *faith* in science. And if we accept that every from of faith is ethical, then all these themes are ethical; in other words, the theme of the antagonistic power between Yin and Yang is an ethical part of the eastern philosophy and the theme of the opposition between phases in percolation is an ethical part of the western science.

Percolation is a twentieth century formulation of an old thing. The last straw that breaks the camel's back is exactly the same as the last jigsaw edge added to a network one step before the onset of percolation (*cf* Tiyapan, 1995, KNT2(ii)). Andrew Lloyd Webber says the same thing with, 'one Rock *n* Roll too many takes its toll and the soul out of you'.‡ But then again, he is also the product of the twentieth century.

§ 4.12 Enlightenment

Both discovery and enlightenment are percolative processes. Like a detective gathering clues suddenly understands the crime he has been investigating, past experiences of a person helps him arrive at the solution. It is likely that much more of these experiences is involved in solving even a seemingly easy problem. The spinning or dancing stars in the algorithm I describe for my solution to the continuum percolation of polygons in § 4.12 can be traced back to my experience in the ancient Thai art of fighting using the swords at the Ayudhàyạ school. Here the ray of the star represents the arm of a person and each side of the polygon the sword he holds in his hand. At the school we learnt to always keep the sword at right angle with the lower arm at all times in order not to hit our friends, since this is no real fighting but a practice. Yet because they correspond to circles with greater radii those parts of the sword away from the hands still hits the swords our friends hold in their hands, and in so doing set off beautiful sparks flying which could prove extremely dangerous if it ever gets into your eyes it scares you more than being hit by the sword. The solution for the flexatube puzzle that I discovered while in Budapest in 1990 is another one of the earlier examples in which a discovery percolates throught after a certain amount of thinking about a problem. I think it takes more than those several days it took me, that is much of my previous experiences with other puzzle counts, which makes it possible for me to solve this particular puzzle. This is not to mention about my long forgotten class in Euclidean geometry which I had to study when I was around twelve years old, *etc.*

The Enlightenment movement in Germany was something different. According to Immanuel Kant(1724–1804) enlightenment is to have the courage to think, as he has told us in an essay (Kant, 1784) of his to have the courage to make use of our understanding.† But the idea is much older than the eighteenth century as the *Bhagavad Gītā*, written written in Vedic Sanskrit and being the sixth book of the epic poem *Mahabharata*, says essentially the same thing. Antithesis to the idea of Kant is the refutation of Karl Popper (1902–1994).

The idea that science develops by leaps and bounds is not new, and examples abundant. Kant may have started the German Enlightenment movement, but Buddha has found a similar thing more than two thousand years earlier. Not only in science, but in other fields for example music, revolutionary discoveries occur all the time that necessarily some of them are old ideas dressed in a new form. Arnold Schoenberg (1874–1951) discovered a revolutionising twelve tone system of harmony which is in fact a mathematical summation of the various scale systems of the ancient Greek. In fact I regard his books (*cf* Schoenberg, 1978) highly not because he was a revolutionising composer but because his understanding in the history of music, which can hardly find an equal. Another

‡ In *The New Starlight Express.*

† Sapere aude! Hab Mut, dich deines Verstandes zu bedienen! ist also der Wahlspruch der Aufklärung.

example is the idea of Immanuel Velikovsky (1895–1979), whose revolutionising book (Velikovsky, 1950), eventhough containing many ancient ideas, is an invaluable asset to the world community. A good discussion on revolution, discontinuity and scientific thoughts was given by Krige (1980).

§ 4.13 Forest fire fighting

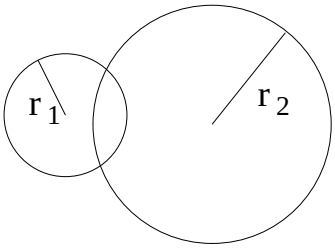
Voronoi graph can be used to aid the decision on the locations of Closed-Circuit Television (CCTV) cameras. CCTVs have proved to be very effective in reducing the number of crimes in Manchester as well as other places within the UK. In order to be effective, it is important to have no large gaps within a control area. Strategically placed, the cameras will give the best cost-effective solution while covering the largest possible area.

Within crucial areas cameras could be positioned in such a way that their ranges of operation intersect without gaps. The points of intersection represent the vertices of the Voronoi regions centred around two or three neighbouring cameras. The design objective is to produce the maximum covered area using the least number of cameras.

Outwards from each crucial area extends space not being covered by cameras. Similarly around other crucial areas surrounding this area also extend such unobserved space. The next step in the design is perhaps to make sure that people walking from one covered area to another are safe. To do this one can consider the whole covered areas as centres of yet another set of Voronoi tessellation orders of unit larger in size. The purpose now is to make sure to position cameras at the positions of vertices of the Voronoi regions of this second set of tessellation.

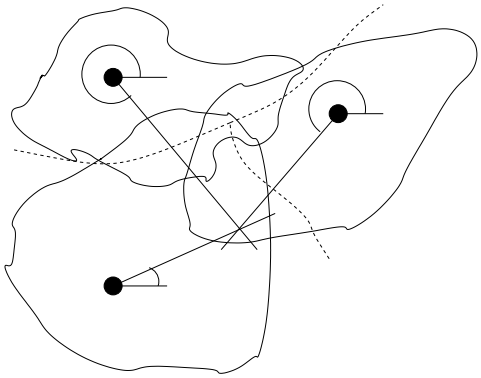
Both the first and the second tessellations will be subjected to geographical constraints. The former would be predominantly influenced by the shape and the position of buildings and surrounding structures, the latter by the paths connecting the covered regions. Here a vertex of these geographically-constrainted Voronoi regions could be a point which is equally far from three covered regions. Another set of positions to be considered for placing CCTVs is one of those points equally far from two covered regions along the shortest path between them.

At a graduate development programme on 16^{th} November 2001 I gave a presentation for ten minutes on a title *division of space*. In it I described the current project of mine as concerning porous media, or division of space in general. The part presented is from a section for miscellaneous applications. Mathematical models of partitionings, for example the Voronoi tessellation pioneered by G. F. Voronoi and G. L. Dirichlet in the 19^{th} century, help towards understanding physical Euclidean and non-Euclidean world. Procedures based on these models help solve many physical and strategical problems. When I was fourteen years of age I was trained in a course for those who volunteer to fight forest fires in the area around the Sudęb mountain, close to where I used to live. Although after the course I never had an opportunity to use it, the knowledge obtained from that course has given me interest in forest fire fighting. To mention but one of the relevant topics contained in that course, one can imagine a forest fire station as being a nucleus of a Voronoi partition of area under the protection of that stations. Here the distance from a station to any point on the forest ground is not a straight line but the time it takes a fire fighter from that station to reach the point in question. This time is affected by the *tortuosity* of the path, in other words the degree of winding, as well as the difficulty of the climb. There normally is an observation tower at each fire station, so we may safely assume that every nucleus is equipped with one. Then the exact location of a fire observed can be located on the map from the intersection of lines of sight drawn from the neighbouring towers towards the direction of the fire. These lines of sight only give the position of the fire on the map. The distance measured along such a line does not necessarily correspond with the Voronoi distance which has a unit of time. The area covered by a tower depends on its height; a higher tower can report a fire occurring at a further distance. The area of covering is circular if the forest is a flat plain. But such a case is rare and in general this area is a distorted circle the degree of distortion of which depends directly on the contour of the land around it.



In Figure 4.15 if h is the height of a tower and r its radius, then $r_1 < r_2$ implies $h_2 > h_1$.

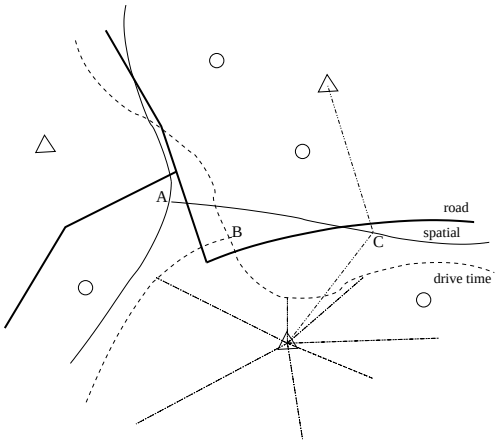
Figure tower. Range of observatory towers.



The dashed lines are boundaries of Voronoi regions, solid boundaries the observation range of the towers, and straight lines from nuclei the lines of sight. In Figure 4.16 the Voronoi boundaries are not straight lines as a result of tortuosity. Observations from only two towers suffice for finding location of the fire, but an observation from the third tower would reduce the probability of an error.

Figure tuos. Tortuous boundaries

I also talked about strategic locations. I once listened to a job presentation by Tesco where I learnt that they use the Geographical Information System to place their new stores in strategical locations. The planning department at Tesco uses what they call the *porcupine diagram* which looks like the rays of light emanating from a star or the quills from a porcupine. These rays radiate from a supermarket and their length represents the drive times required to get to the store from the other end of the line. Factors affecting the locating of new stores include for example drive time, competition, logistic, parking, and traffic.



The porcupine diatram superimposed on a map. Triangles are Tesco supermarket stores while circles are those belonging to their competitors. The spatial boundary of the Voronoi diagram is in general different from its drive time boundary, as shown in Figure 4.17 where broken boundary lines represent Voronoi partitions when the distance is the drive time while solid boundary lines that when the distance is the spatial distance.

Figure fstra. Porcupine diatram superimposed on a map

A similar approach to the above can be applied to help locate the Closed Circuit TeleVision cameras in a strattegic location. Suppose that now the triangles in Figure 4.17 are important places, for examle business centres, schools, or universities; let us call these *sensitive areas*. Then security cameras should be placed at both points A and B as well as at the sensitive places. Here the solid boundary lines meeting at the point A represent the Voronoi boundary for walk time instead of the actual spatial distance. Moreover, a camera should be placed at the point C which is the point where the shortest walkway linking two neighbouring sensitive areas meets with the Voronoi boundary of the walk time, in other words a point which lies midway along the shortest path that joins nextdoor sensitive areas. The reason for this is that these points are likely to be frequented by people going from one sensitive area to another, therefore they are prospectous for robberies. On top of that if a robber wants to flee from one sensitive area to another his sensible choice would pass through C while A with its connecting three different routes would be a clever choice to choose iconfound a pursuer, or likewise B if he is driving.

I had been first to the police headquarter and then to the city council office because I wanted to know whether security cameras have been placed at locations analogous to these points A, B, and C, but has been told that the information I asked for is a sensitive one. The answer is understandable because at the time President Bush was trying to get his hand on Bin Laden and Prime minister Blair had just declared Britain at war and alert. However this was not good for my theory since I had no way of knowing whether this approach is a new one, or whether it could be used to improve the security in the city even further. I hope that most of these strategic locations A, B and C have already been covered by security cameras even if the decision to do so had been arrived at by some other theories different from the one I suggest. Unlike in the case of sensitive places, the cost of the cameras to be placed at these strategic places may have to be the responsibility of the city council because their importance is not immediately obvious owing to their locations. In contrast, most schools, universities or businesses should be willing to help with the cost of such installation from which they would directly benefit.

On one slide I explain how one can divide transformations into affine and non-affine ones. The affine transformations are governed by the equations $x' = ax + by + p$ and $y' = cx + dy + q$. As a conclusion, mathematical models are beautiful because they give us understanding, and they are useful because they stem ideas and prpocedures necessary for solving problems. The presentation has gone from forest fire fighting to crime prevention without my knowing it, so the future works are to collect real data and practices, to analyse and compare with the theory and simulations, and then to implement and monitor the operations which I think unlikely to happen because it is impossible

for me to become a member of the Greater Manchester police force. Dr. Jim Boran who was present gave *good*'s for project understanding, overview and structure, summary, examples, use of colour, connection with drawings, enthusiasm, pointing at screen, and familiarity with the material. He wrote that the explanations were clear, the content technical but with good personal view, but that I looked at the screen too often and at times was difficult to hear. Bigger lables are suggested.

Naval forts are positioned with an idea similar to forest fire stations. Each fort forms a circle of effective defence around itself. These circles form a chain like beads on a necklace. One example (*cf* Petersburg, 2001) of this are the forts of Kronstadt near Kotlin Island in the Finnish Gulf, 20 kilometres from Saint Petersburg, which have been used as a base for the Baltic Fleet and is now controlled by the Russian Navy. Here seventeen forts had been built between 1704 and 1896 which form a guarding barrier for the city of Saint Petersburg. Examples of these forts are the Totleben Fort, founded 1886–1913, and the Alexander I fort, founded 1836.

In the study of traffic in § 7.1 we consider the percolation of roads, which are edges of the structure. We can also use the same structure for the study of fire protection and evacuation in cities, in cases of disaster, but now it is instead the percolation of cells that interests us. Roads form effective fire barriers as well as divide the conflagration into partitions. By carefully managing these, for example putting more efforts on the key or critical partitions, the flames can be contained more quickly and the damage minimised. These zones are also important for evacuation planning, for example that made by the Greater Manchester Police. In Tokyo where there is always a fear of big earthquake, informations are given out in plenty to tell people what to do in case such a disaster as the Great Kanto Earthquakewhich occurred on 1^{st} September 1923 should struck. School grounds are usually assigned as evacuation centres. And as the distance one has to walk to get to one of these is crucial for one's safety, the evacuation zones assigned by the city councils there naturally divide the city into domains that approximate the Voronoi tessellation. In 1998 when I was in Tokyo the fear reached its height, since it was generally believed that big earthquakes in Tokyo occur at a period of approximately seventy-five years.

§ 4.14 Fractals and percolation

In doing simulations on filtering membranes, one usually assume that the blocking particles are all of the same size. Or one may apply more intuition and experience, and say that particles should have a normal distribution. Any mineral engineer will be able to tell that it is normally the case that the distribution of aggregates according to their sizes is normally normal. But one must keep in mind that the normal distribution here is by weight not number of particles. Therefore it is unlikely, except in a few special cases, that simulations which assume a normal number distribution of blocking particles will represent the real process.

A pile of aggregates with a normal distribution is likely to have a fractal number distribution (*cf* Liebovitch and Scheurle, 2000). Such fractal distribution is hyperbolic towards the lower size ranges. But the overall distribution is more complicated, that is to say, a mixture of normal- and hyperbolic distribution.

So far we have only considered percolation by competition between two phases. We have looked at networks in a cubic box and have assumed that the network is initially available in its entirety. But what if one has a competition of three phases or more? Or what happens when there are only parts of the network from the start? The picture of this can not be clearer than that of politicians arguing for votes, or the opinion polls of people across the social cross-section. Lustick and Miodownik (2000) consider this problem in the context of politics where the agreement clusters necessarily appear in various colours not merely black and white.

Applying this to percolation and a whole new dimension opens up for investigation. One possible approach is to consider only an arbitrary part of the network within our cubic box. Clearly this is only meaningful if the part being considered is itself a percolating cluster. Carry this one step further and we can have percolation within percolation, when each cluster within the network is itself subject to another percolative process, threatening to destroy it for instance. Such a dynamic scenario is a yet untread water which should open up much ground for investigation.

In the early days of the theory, literature on percolation only concerned itself with fluid wetting or blockig bonds or sites in the kind of situation that is said to be dual to the diffusion problem (Broadbent and Hammersley, 1957; Hammersley, 1957, 1961). The subject generally looked at self-avoiding walks and a measure of connectivity, the connective constant. From then on the appearance of the subject has somewhat changed, and it has become connected to Physics and fractals. The reason why there are very few works on percolation of a random network like the Voronoi tessellation

is probably because there seems to be no need for such study, since the applications in Physics have told us that the regular lattices can represent the noncrystalline structure of the metal well. Moreover the only tools we have now, namely the Hamiltonian and power series method can not be applied in its present form to random tessellations, since there are no modulus relationships among the vertices.

5

§ 5. Porous media

Patrik *et al* (1999) study the propagation in random Delaunay lattice. Particle on arriving at a site in the lattice deflects over the largest possible angle to either the right or the left, depending on the right- or the left nature of the scatter site. After the particle has passed through the site, the latter goes into the reverse state. When a similar study is done on the triangular lattice, the entire trajectory quickly becomes confined to a particular strip which is bounded by two adjacent parallel lines of the lattice. They explain the propagation as being due to a blocking mechanism which prevents the particle from moving in a direction opposite to the propagation direction for more than a few steps.

The flow of fluid in porous media follows the Darcy's Law which states that the rate of flow through such a medium is proportional to the potential energy gradient within that fluid. The constant of proportionality is the hydraulic conductivity, which is a property of both the porous medium and the fluid moving through it, $v = Q/A = -k\mathrm{d}h/\mathrm{d}l$.

The average velocity over the entire cross section is called the superficial velocity, $v_s = Q/A = \dot{m}/\rho A$. The velocity that is based on the actual open space within the porous media is called the interstitial velocity, $v_n = Q/(\varepsilon A) = \dot{m}/\varepsilon\rho A$, where ε is the porosity, $\varepsilon = V_v/V_s = (V_t - V_s)/V_t$, V_v, V_s and V_t are respectively the void, solid and total volume.

The definition of porous media is different from that of porous materials. A porous medium is a medium through which other substance may pass, whereas a porous material merely means some certain kind of material the internal structure of which is filled with pores. Most of porous materials can be used as porous media. All of them are useful because of their internal structure, and though we do not need to know this to be able to use them, we do have to understand it if we want to use them efficiently, or if we want to improve upon some of their particular properties.

The outer bark of Quercus suber L., for example, has a property that is ideal for its commercial use as cork. Its material property is such that it neither shrink nor expand when stretched or compressed. This makes it an excellent insulator of both heat and sound as well as good for damping vibration (*cf* Ashby, 1990). It has high frictional coefficient, and is both impervious to liquids and chemically stable. These properties make it ideal as wine corks. By understanding its internal structure and mechanism that makes it stay in the same shape when experiencing external forces, scientists have succeeded in making a synthetic material that shrink, instead of expand radially when pressed. When such material is used as cork, it can be easily pushed inside the neck of a bottle. When we release the force it will expand to its full size and fit snuggly where we placed it.

Pores introduced into a fabric can also increase its commercial value by giving it a lighter weight, higher heat retention rate, which makes it ideal for both ski and snow-board wearer and casual wear (JETRO, 2001).

§ 5.1 Medical science

Many diseases can be best thought of as related to percolation when a threshold exists passing the point of which it is difficult to return or to get back to normal. For example a cut may take several weeks to completely heal, but muscular strains could take several years and even then they usually leave a weak spot and thus can not be considered as completely healed. This means that despite the turn over of cells within the fibers of the muscle, the strain still prevail. In other words it has buried itself into the structure of the fiber and no longer is related merely with the individual cells. Not only muscular fibers, but also the brain and the nervous system may be viewed as a kind of networks. Therefore it is not difficult to see many nervous or brain diseases as percolated phenomena. Madness and its onset is an easy example. Deafness, either gradual or sudden, is another one. The difference between a sudden and a gradual onsets of deafness lies in the position of the percolation threshold on the timescale. In the former the network percolates after a period of hidden graduation or deterioration in the condition of the inbuilt sound detector, while in the latter it percolates before the graduation down towards the deafness. Most books on otolgy talk about sudden onsets of deafness, normally overnight. The reason they are much talked about is more likely to be because of their tragic nature than because they are common. It is interesting to note that the word *otology* means the study of the ears, *-logy* is suffix which comes from the Greek word *logos,* meaning *the study of,* while *ot* or *ōt* means *ear.* It is probably a coincidence rather than a liguistic relationship between the two languages that *oto* is the Japanese word for *sound.*

I should have something to say about the Menière disease because both my previous general physician Dr. Sreedharan and the present one Dr. Chan think that it is what I have. I do not quite agree with them, but if both of them were right and I am wrong then Menière disease results from the accumulation of certain things which then percolates, and I think I know what these things are. This would have been rather a medical breakthrough because all medical textbooks that I have come across anonymously agree that this is a disorder the cause of which is excluded from its definition. In other words all we know about is what its symptoms are, nobody knows what causes them. What I maintain is this; provided that both of my doctors diagnosed correctly then the cause of it must be percolation. Having said that, I do not think that they did diagnose right, for eventhough I have both the pressure in the ear and the tinnitus, I never have vertigo. It is true that I have some experience of syncope which is even worse than vertigo, but they are not the same thing. Physicians, however, insist that you must have vertigo no matter if you say you don't, because then they would know that you have the thing called Menière disease the definition of which they do not know anyway! The disease is named after a french physician Prosper Menière who first described it in 1861.

Tinnitus is the constant rumbling or ringing in the ears. The sound can be real and sometimes can be heard by others. Many things can cause it, for example (*cf* Andrews, 1997) excess noise level, alcohol, allergies, antibiotics, blood glucose or blood pressure swing, circulation changes, cranial or jaw-joint misalignment, fever, inflammation, temporomandibular jaw joint diseases, tumours on acoustic nerve. It can also be caused by metallic – aluminium, lead, or mercury – poison. If the cause of it is a prolonged or excessive level of sound, then the process would be similar to that which causes, for instance, the repetitive strain injury.

I will now explain why it should have something to do with percolation. First, I think it is sound or the excess of it which causes the percolation. Next let me give some facts which lead me to the conclusion above. Around the turn of this millenium a friend of mine bought two concert tickets and invited me to join her. Seldom been to a concert, I was delighted. And having noticed that the program includes a piano recital of one movement from a sonata, probably the fifth, by Scriabin, I decided that I would do my homework in order to gain as much as possible from the concert. At that time I had recently bought the music, a complete sonatas by Alexander Scriabin. They are a total of ten sonatas, and I sat at the piano and read them one by one, day after day. Ten hours a day at the piano was sufficient to give me the obvious sign of an ear problem. For a few days I could not hear anything clearly, and could hardly speak normally. The second sign came not long after I had been to a Community Action conference held in Liverpool. The conference was something new and was very interesting, but I discovered there that young volunteers enjoy too loud a music on the dance floor at night to the cost of my ears. Again the following few days gave me the most alarming feeling, I hardly could hear anything, and whatever sound I heard seemed like coming from the outside to me inside a sealed tank. This together with the experience of having lived in Bangkok, the city where noise polution ranks number one, and that of having listened to loud music myself when I was young, have combined to reach the limitting threshold. As I am writing now I still have yet to recover from the second trauma to the ears mentioned. There is a saying that

the third time pays for all (*cf* Tolkien, 1955), and so I hope that the third time shall never come considering that the second one has already been this bad and I am in doubt whether I will be able to recover from it. But to me it is clear that this has got something to do with a limitting threshold and thus percolation.

In the case of Beethoven, who also had an ear problem which results in a total deafness, most modern physicians think that the cause was multiple sclerosis, which I also disagree. As a great pianist and a conductor he was exposed to much sound, so I suspect that the latter is also the culprit in his case. It seems an irony that for such people, as well as for myself, music is life and music hurts.

Some thinks of multiple personalities as the fragmentation of the mind or brain (*cf* Keyes, 1999) which causes a person to identify himself as different personalities each of which has its own memory separate from the others. There is a transfer of control at the switching from one personality to another, and all the personalities could use the same body in turn. It is more likely that instead of fragmentation the various personalities are different phases coexisting in the same network of the mind. Like a polyglot or a normal person switching phases among different languages he knows, a person with multiple personalities unconsciously switches himself from one self of his to another. Multiple personalities often report head voices of one self talking to another. These head voices are distinct conversations and thus are different from those experienced by people with schizophrenia. The various personalities have different Electro-Encephalograms, Intelligent Quotients, and psychological test results. This altered state phenomena is normally cured by fusing the personalities together, but some say that the this is not necessary as long as all the personalities are aware of one another. Similar to multiple personalities a person is often said to have good and bad sides, or in some cases selves (*cf* Stevenson, 1886).

We still do not understand much about sleep, that state where the mind finds itself in phases which are different from the awaking self. No one seems to know why it is neccessary that we sleep at all, but all animals sleep, or die if they are deprived of it. Dolphins sleep one hemisphere of the brain at a time. Sleep and wakefulness are two different phases of the mind, and the sleep itself may contain more than one phase. The transitions back and forth among these various phases are the same as the changes of phases characteristic to percolation. A person normally has several dreams while he sleeps, but only the immediate one from which he is wakened up is remembered. This may mean that during sleep the mind is switched among various phases, or it may mean that in sleep the mind only has a short-term memory.

Subconscious mind can sometimes come to the front. This often happens when the conscious mind becomes weak. When a man becomes drunk, for instance, he may be able to find his way home only to wake up the following morning, sober and wondering how he has got there. This is because in the drunken state his mind becomes muddled up by the effect of alcohol that the subconscious mind took control to lead him home unconsciously. Multiple personality represents the fragmentation of the subconscious mind into virtual clusters each of which then alternately comes to the front to parade as a person. These clusters are probably superimposed in three or higher dimensions, without overlapping one another, in a way similar to the intertwining of clusters in a three-dimensional Voronoi network. The change of personality in this case is then a transfer from one phase to another. The first and most important fragmentation which leads towards a multiple personality seems to happen only in children at around the age of three, when the mind is in the forming and the language of the mind is predominantly geometrical, while later and minor fragmentations are possible much later afterward. Our minds during sleep seem to be another layer separate from and exists below both the conscious and all fragments of the subconscious mind, as can be seen by the fact that a person with multiple personalities also needs to sleep. Some of the most fearsome and the least understood sicknesses like Parkinsonism and Encephalitis Lethargica, or the sleeping sickness, affects only the conscious part of the mind, not the higher faculties of the unconscious part (*cf* Sacks, 1973). One of the effects of the medicine laevo-dihydroxyphenylalanine, or L-DOPA, seems to be that of letting the subconscious mind take over to motor functions. The condition of the patients usually improves quickly soon after first put on the medicine, then overshoots a stable condition and shows adverse effects of the medication which are no less gruesome than the symptoms before starting to take the drug. These counter symptoms probably occurs in Control Systems as an overshoot in the output in response to a step function which represents the usual constant daily dosage used. Because such undesirable symptom occurs in most cases, dosage should either be reduced before the overshoot occurs in accordance with the empirical time constant of the response, or increased very gradually from zero at the beginning in order to reach the steady state be

reached without an overshoot, as an analogous to an over- or critically damped control scheme. One of the most intriguing effects of too much use of L-DOPA is that primitive, sometimes subhuman, behaviours could be brought to the front while the patient himself can only stand back as a witness to his own self without being able to do anything to stop them.

In the art and science of Neurology there is no lack of phase transition phenomena. So much so that symptoms normally come in pairs of opposing poles, for example aboulia – hyperboulia, automatism – command-negativism, bulimia – anorexia, and perseveration – block. Within these pairs the poles may switches suddenly as can be seen in kinexia paradoxa, the explosive transition between hyperkinesia and akinesia. Somewhere between the two poles lies our normal conditions. In many of post-encephalitic patients taking L-DOPA, this middle ground which starts off very wide can rapidly shrink until the patients find themselves balancing on the edge of a knife and it becomes impossible to calibrate the drug to the right amount.

§ 5.2 Zeolites

Zeolite is the name given to a group of minerals with a porous structure. It is used as molecular sieves as well as in many chemical engineering processes. It may be an important ingredient responsible in lubricating the fault lines in earth crusts, thus making earthquakes less severe. Evan *et al* (1995) found foliated cataclasite and ultracataclasite in the San Andreas faults. These are composed of clay and zeolite. In the case of the ultracataclasite, there are fragments of 20–100 μm diameter feldspar and quartz embedded in a matrix of clay and zeolite which has grains of sizes smaller than 10 μm.

The chemical composition of zeolites is that of sand, *i.e.* aluminosilicate. Their general formula is (Dana and Dana, 1997)

$$(\mathrm{Na, Ca, Ba})_{(1-2)}\,(\mathrm{Al, Si})_5\mathrm{O}_{10} \cdot \mathrm{nH_2O},$$

or, as given by Gottardi and Galli (1985),

$$(\mathrm{Li, Na, K})(\mathrm{Mg, Ca, Sr, Ba})_{\mathrm{d}}[\mathrm{Al}_{(\mathrm{a+2d})}\mathrm{Si}_{\mathrm{n-(a+2d)}}\mathrm{O}_{2\mathrm{n}}] \cdot \mathrm{mH_2O}.$$

Normally $m \leq n$. They differ from sand in that they have large internal cavities. These cavities increase their internal surface area as a result of which they can be used as catalyst or molecular sieves with both shape- and size-selectivity. One example is the use of zeolite as sieve to separate iso octane, which has high antiknock property, and iso pentane from octane and pentane. Zeolites can not withstand high temperature because they rapidly lose water. Eventhough this process is reversible, their molecules collapse at temperature higher than 600°C. Around 41 types of zeolite occur in nature. Linde Division of the Union Carbide Corporation produced the first synthetic zeolites in 1950s.

The name 'zeolite' was coined by A. F. Cronstedt in 1756 from the Greek *zein*, to boil, and *lithos*, stone. The structural units of zeolite comprises of the primary building units of silicates in the form of XO_4 tetrahedra, where X is mainly Si. Other basic units are the chain of fibrous zeolites, the singly connected 4-ring chain, the doubly connected 4-ring chain, the 6-ring either single or double, the hexagonal sheet with handles and the heulandite unit.

§ 5.3 Crystalisation

In the diffusion theory of crystal growth, the overall rate of crystallisation is determined by the rates of two processes occuring one after the other, that is diffusion of solute from the bulk solution to the interface between solution and the crystal, followed by integration of solute atoms into the crystal lattice. The rate of diffusion of the former is $R_g = k'_d(c - c_i)$, while the rate of surface reaction or integration is $R_g = k_r(c_i - c')^n$, where c_i is the interfacial concentration. The value of the diffusion mass transfer coefficients k'_d when estimated from the growth is considerably different from the same value that is obtained from the dissolution experiment. But if we assumed that they are equal, then $R_g = k_r(\Delta c - R_g/k_d)^n$, which, when $n = 1$ gives $R_g = K\Delta c^n$ where $1/K = 1/k_d + 1/k_r$, and when $n = 2$, $R_g = k_d\left[(1 + k_d/(2k_r\Delta c)) - [(1 - k_d/(2k_r\Delta c)^2) - 1]^{1/2}\right]\Delta c.$, but there is no general solution for all n's (*cf* Garside and Mullin, 1968).

Particles in a crystalliser are kept in suspension by a stirrer or a pump. Lim *et al* (1999) study the effect of the attrition when crystals encounter with high speed impellers inside these apparatuses. When the impact energy exceeds the crystal strength, crystals fracture which gives rise to a particle size distribution as a result. Impact can occur at faces, edges or corners of crystals, but in the attrition model only the contact of a crystal corner with another flat, much harder object, for instance the steel impeller, was considered. Furthermore, crystals are assumed to have the shape of a cube or other polyhedra, but in the model the crystal faces forming the contacting corner are replaced by a cone having an included angle of 120°. In the vicinity of impellers the flow is turbulent and crystals travel in a random manner. Repeated attritions reduce the crystal size, assuming no competing effect of crystallisation. They assume a normal volume shape factor distribution and proceed to simulate according to a procedure recaptured here as Algorithm 5.1. The input to the algorithm is the impact energy W_p; the outputs are $w_f(J)$ and $c(J)$. The hardness of a solid is its resistance to local plastic deformation. The contact pressure of a plastically deformed cone is assumed to be the same as the Vickers hardness H_v. An isotropic material has two independent elastic constants, namely the shear modulus μ and the Poisson's ratio ν. All the other constants can be determined from these two quantities. Here α is the volume shape factor, Γ the fracture resistance, K_r the efficiency of stress field created by the impact between the crystal and the impeller, κ stress field parameter, $\kappa = 5$, μ_v quasi-isotropic shear modulus, N the total number of fragments, W_p impact energy, H dynamic hardness, $L_{\min}$ and $L_{\max}$ respectively the minimum and maximum size of a fragment, a characteristic size of the plastic zone, r the distance from the peak of the cone, $r_{\max}$ the maximum distance from the peak of the cone to the newly created surface and J the particle size classes. The values of H_v, μ_v and W_c used were that of magnesium sulphate heptahydrate and potash alum; r_1 and r_2 are random numbers.

Algorithm 5.1 *Fracture by attrition, Lim et al (1999).*

for $i = 1$ to 50 **do**
  $\alpha \leftarrow r_1$;
  $\Gamma/K_r \leftarrow \kappa W_c^{1/3} H^{5/3}/(5.2\mu)$
  $N \leftarrow 7 \times 10^{-4} W_p H^5 K_r^3/(\alpha\mu^3\Gamma^3)$;
  for $j = 1$ to N **do**
    $L_{\min} \leftarrow 32\mu\Gamma/(3K_r H^2)$;
    $L_{\max} \leftarrow r_{\max}/2$;
    calculate $r_{\max}$;
    $r \leftarrow \exp\left[(13\log a - \log r_2)/13\right]$;
    $L \leftarrow 3\mu\Gamma r^4/(W_p^{4/3} H^{2/3} K r)$;
    find J corresponding to L;
    $w \leftarrow w + L^3$;
    $w(J) \leftarrow w(J) + L^3$;
    $c(J) \leftarrow c(J) + 1$;
  endfor
endfor
$w_f(J) \leftarrow w(J)/w$; □

§ 5.4 Fluid flow within networks

Flow of viscous fluids through networks of geometrical objects is an important topic in various disciplines of engineering. Happel (1959) studies two cases of the flow of viscous fluid relative to arrays of cylinders, one parallel while the other perpendicular to the cylinders. This is found in practice as the flow through a bundle of heat exchanger tubes.

The flow pattern within a void of a porous media follows the equations of flows with vorticity, in 2-d

$$\frac{\partial^2\psi}{\partial r^2} + \frac{1}{r}\frac{\partial\psi}{\partial r} + \frac{1}{r^2}\frac{\partial^2\psi}{\partial\theta^2} = \zeta,$$

and in the axisymmetric case

$$\frac{\partial}{r\partial r}\frac{\partial\psi}{\sin\theta\partial r} + \frac{\partial}{r^3\partial\theta}\frac{\partial\psi}{\sin\theta\partial\theta} = \zeta$$

(*cf* Rowe, 1965). Or equivalently to both equations,

$$\zeta = \frac{1}{r}\left(\frac{\partial(rv_\theta)}{\partial r} - \frac{\partial v_r}{\partial\theta}\right),$$

when $v_\theta = \partial\psi/\partial r$ and $v_r = -\partial\psi/(r\partial\theta)$ in 2-d, and $v_\theta = \partial\psi/(r\sin\theta\partial r)$ and $v_r = -\partial\psi/(r^2\sin\theta\partial\theta)$ in the axisymmetric case. The vorticity, ζ, can be used as boundary conditions, for instance $\zeta|_{r=0} = -\partial v_r/a\partial\theta$ or $\zeta|_{r=a} = \partial v_r/a\partial\theta$. Or it can be used within the hole, for example $\zeta = k\sin\theta$ which gives the solution $\psi = (c_1 r^{-1} + c_3 r + c_4 r^2)\sin\theta$, or $\psi = (c_3 r + c_4 r^2)\sin\theta\sin\theta$ if $c_1 = 0$. This solution is a part of the general polynomial for the stream function,

$$\psi = (\cdots c_0 r^{-2} + c_1 r^{-1} + c_2 + c_3 r + c_4 r^2 + \cdots)\sin\theta.$$

§ 5.5 Material science

A chiral is a group of points, or geometrical figure, whose mirror image can not be brought to coincide with itself. Achiral is the antonym of chiral. A chiral object is an object which fails to be achiral. Chirality is a purely geometrical property since since all the operations involved, namely the plane symmetry of the reflection, the rotation, and the translation, are all isometries. Chiral is used in the studies of molecules and knots.

The Poisson's ratio, ν, is the ratio between the transverse contraction strain and the longitudinal extension strain, that is $\nu = -\frac{\varepsilon_t}{\varepsilon_l}$. The theoretical value of the Poisson's ratio can range from -1 to 0.5, but for normal materials it is generally positive. At $\nu = 0.5$ the bulk modulus is much greater than the shear modulus and the material is incompressible, while at $\nu = -1$ the opposite is true and the material is very tough and highly compressible. The bulk modulus B, and the shear modulus G are relatated to each other by the equation $B = \frac{2G(1+\nu)}{1-2\nu}$. For rubber this value is 0.5, for aluminium 0.33, while for cork it is approximately zero. Materials with a negative Poisson's ratio have been found whose structure is re-entrant (Lakes, 1987). Applications of such materials include robust shock absorbing material, fasteners, and stoppers of the wine bottles.

Chemical reactions can be thought of as phase changes in percolation. For example the extraction curves for Sb shown in § E.1. These curves represent an s-shape starting from one phase, represented by 0% Sb, to another at the maximum per cent extraction where it saturates. All the per cent extraction graphs shown in § E.1 show them with high slope at $t = 0$. This is only for the convenience of drawing, since it is difficult to draw curves with an s shape smoothly on the computer. Also, the data obtained from the experiments do not extend to the time immediately following $t = 0$. Extracting the solution using pipette takes some time to do, and therefore it has not been possible to prove experimentally whether the extraction yields start off with zero slope at $t = 0$ or not, eventhough one might conjecture that this is likely to be the case if one considers the extraction yield in the leach solution as a developing phase in the continuum of the solution.

Particle sizes come in a variety of definitions. Svarovsky (1977) divides them into three groups, namely definitions by equivalent sphere diameters, by equivalent circle diameters and by statistical diameters. These are listed together in Table 5.1.

diameter	**criterion**
volume diameter	equivalent volume of sphere
surface diameter	equivalent surface of sphere
surface volume diameter	surface to volume of sphere
drag diameter	resistance to motion of sphere in the same fluid at the same velocity
free-falling diameter	free-falling speed of sphere, same fluid and particle density
Stoke's diameter	free-falling speed of sphere if Stoke's Law is used ($Re < 0.2$)
sieve diameter	diameter of sphere passing through the same square aperture
projected area diameter	projected area of a circle, the particle resting in a stable position
projected area diameter	projected area of a circle if the particle is randomly oriented
perimeter diameter	perimeter of the outline of a circle
Feret's diameter	diatance between two tangents on opposite sides of the particle
Martin's diameter	length of the line which bisects the image of the particle
shear diameter	particle width obtained using an image shearing eyepiece
maximum chord diameter	maximum length of a line limited by the contour of the particle

Table 5.1 *Particle size definitions.*

Particle size distribution comes in four types, distribution by number $f_n(x)$, by length $f_l(x)$, by surface $f_s(x)$ and by mass or volume $f_m(x)$. Conversions among them are done by $f_l(x) = k_1 x f_n(x)$, $f_s(x) = k_2 x^2 f_n(x)$ or $f_m(x) = k_3 x^3 f_n(x)$. Conversion is only possible if we know the shape factor's dependence on particle size, because k_i often contain a shape factor. The distribution frequency is by definition $\int_0^\infty f(x)\mathrm{d}x = 1$.

$$(m, \mathrm{d}f(x)/\mathrm{d}x, Q) \xrightarrow{\text{feed}} \text{separator} \xrightarrow{\text{overflow}} (m_f, \mathrm{d}f_f/\mathrm{d}x, O)$$
$$\text{separator} \downarrow \text{underflow}$$
$$(m_c, \mathrm{d}f_c/\mathrm{d}x, U)$$

Figure 5.1 *Schematic diagram of a separator.*

Small particles tend to flocculate with one another. This makes it impossible to do experiments with very fine powder. Therefore any experiment which claims too fine a size as a control parameter,

for instance in the micro metre range, must bear in mind that particles of such small size ranges tend to cluster into hard agglomerates bound by strong chemical bonds, for example from a previous chemical or thermal treatment, or into soft agglomerates by van der Waals attraction or by capillary forces. In ceremic making, this results in nonuniform packing during forming which leads to large voids or flaws after thermal processing. As one expert put it, you can not have powders in the size range of microns because they will disappear instantly into the air. One way to deal with them is to mix them into slurry.

Powder is also subject to impurities from the earliest stage of its life, that is during the comminution, which can be done by various methods, for example mortar and pestle, ball milling, jaw crushers or crushing rollers. The impurity comes from abrasion pf the grinding media, which can be effectively avoided in the case of the jet mill, where particles are driven by air streams in opposing directions to collide with one another among themselves. Powder can be characterised by its size distribution, surface area, shape, composition and crystal structure.

Particle size distribution can be determined by means of Stoke's law, screening and microscopy. Stoke's law states that the steady state velocity of a spherical particle travelling through a fluid is $v_r = 2(\rho_p - \rho_f)ar^2/9\eta$, where a can be due to gravity, in which case $a = 9.8$ m^2/s, or alternately due to centrifugal force, $a = \omega^2 x$. The time required fro a spherical particle of of radius r to travel a depth x is $t_r = x/v_r$. No particles of redius larger than r will be found at depths less than x at time t_r. The fraction extracted mass at short time intervals in an experiment using a graduated cylinder is constant, $\pi R^2 \int_{-k}^{0} f_o \mathrm{d}x = m$.

The equivalent spherical radius is the radius of a solid sphere that has the same steady state velocity in the slurry as the particle. The volume distribution function is $f_v(r) = \mathrm{d}V/(V_\mathrm{T}\mathrm{d}r)$. The number of particles in the powder having radius r is $N(r) = 3V_\mathrm{T} f_v(r)/4\pi r^3$, the mode of the distribution is the peakk and the mean radius is $\int_0^\infty rN(r)\mathrm{d}r/\int_0^\infty N(r)\mathrm{d}r$.

When finding particle size distribution using screening methods, sieves are arranged in a stack with their apertures decreasing from top to bottom. Mass fractions retained on each of the screens is $f_i = m_i/m_\mathrm{T}$, which can then be converted to $N(r)$.

Microscopy methods use a variety of microscopes, for instance optical, SEM or TEM, depending on the magnification required. Measuring the dimensions and determining the shapes of the particles can be done manually with a ruler, or by using an image analyser.

Powder surface area is determined by gas adsorption methods. The specific surface area, *i.e.* surface area per unit mass, can be found by $s = N\sigma V_m/V_o$, where N is Avogadro's number, σ_0 the effective cross sectional area of the adsorbate molecule, V_m the volume per gram of a monolayer at STP and V_0 the STP number, 2.24×10^4 cm^3/mole.

Monolayer adsorption is used when molecules strongly interact with a surface to in a uniform monolayer. The Langmuir equation is $\theta = kP/(1 + kP)$, where $\theta = V/V_m$ is the fraction of surface covered and K a constant. Then $P/V = 1/(KV_m) + P/V_m$ and the Langmuir isoterms are plots between V and P/P_0, P_0 being the saturated vapour pressure. For multilayer adsorption, $x/[V(1-x)] = 1/(cV_m) + x(c-1)/cV_m$, where $x = P/P_0$.

Permeametry measures the flow of gas under a pressure head through a packed bed of particles. The Darcy's law of laminar flow through porous medium is $u = \mathrm{d}V/(A\mathrm{d}t) = K\Delta P/\ell$. The Poiseuille equation for a fluid flowing through a tube of radius r is $Q = r^4\Delta P/(8\eta\ell)$, where η is the viscosity. It is also written as $h/L = 32\eta v/(gd^2)$, where h/L is the headloss per unit length, η the kinematic viscosity (m^{2}s^{-1}), $\eta = \mu/\rho_w$, ρ_w mass density of the water (kg $\cdot$ m^{-3}), μ absolute viscosity of water (kg(ms)$^{-1}$) and d the diameter of the particles. The hydraulic radius is the ratio between the wetted area and wetted perimeter, $r = (\pi d^2/4)/(\pi d) = d/4$. Therefore, $h/L = 2\eta v/(gr^2)$. The number of particles is $n = V_\mathrm{T}/V_i$, and the total wetted surface area of solids is $A_\mathrm{T} = nA_i$. Therefore $A_\mathrm{T} = (1-\varepsilon)(\pi d^2)/(\pi d^3/6) = 6(1-\varepsilon)/d$. Then, replacing the factor 6 with the shape factor s, $r = \varepsilon d/[s(1-\varepsilon)]$. Because $v_p A_p = v_f A_f/R$, where p and f means pore and filter surface respectively, R is a constant which accounts for friction loss, and also $J = 1/R$, therefore we have $h/L = 2\eta v_p/(gr^2) = J\eta(1-\varepsilon)^2 v_f s^2/(g\varepsilon^3 d^2)$.

Kozeny used the ratio $V/A = d/4$ for a cylinder and the porosity $p = V_v/(V_v + V_s)$ and gavethe equivalent cylindrical diameter for a packed bed as $d_e = 4pV_s/[A(1-p)]$. Then the average velocity through the channels is $u_p = \mathrm{d}v_p/\mathrm{d}t = [(V_s^2\Delta P)/(2A^2\eta l)(p/(1-p))^2]$, where l is the channel length. The Carman-Kozeny equation accounts for the tortuosity of the channels in the bed, $s^2 = 2.5p^3\Delta P/(2\rho\eta l v(1-p)^2)$, where $V_s = m/p$, the specific area $s = A/m$ and ρ the solid density of the powder. The determination of the phases and composition of a powder is done by x-ray powder diffraction, thermogravimetric analysis and differential thermal analysis.

Wet methods are preferred when preparing material before firing in order to control agglomeration during forming, for otherwise when the particles are aglomerated the narrowest particle size distribution becomes of little or no help. In wet forming, solid particles are suspended within a liquid the chemistry of which can be adjusted to make them mutually repulsive or mutually attractive, for respectively deflocculation and flocculation. Stabilising makes particles mutually repulsive either by adsorbing polymer chains on to the particles in steric stabilisation or by putting charged ions or polar molecules on the particle surface in electrostatic stabilisation. In steric stabilisation, one end of the long polymer chains, which is hydro- or lyophobic, is adsorbed on the particle while the remaining end, which is hydro- or lyophilic, extends in to the liquid.

Electrostatic stabilisation results in diffusive double layer built around particles. Zeta potential measures the repulsive potential between particles travelling in a medium. It defines the electrical potential in the double layer at the surface of shear slippage when the particle is forced to travel through the fluid when an electric field is applied. The composition and thickness of the double layer can be changed by adjusting the pH of the suspension. The point where the pH gives a zero zeta potential is called the isoelectric point. It is a point where spontaneous agglomeration occurs. The electrophoretic mobility $\nu_e = u(E)/E = \zeta\epsilon_r\epsilon_0/(f_h\eta)$, where u is a steady state velocity, f_h a constant determined by the dispersing medium and the particle size and $1 \leq f_h \leq 3/2$.

§ 5.6 Forces between particles

Essential to the simulation of stochastic models of particles is the consideration of forces acting on each particle (*cf* Schumacher, 1996). Since in theory the forces acting between two particles extend their effects to infinity, one has to make approximations. The degree of justification to these aproximations depends on how the force in question vary with distance.

In the case of the Coulomb force where the potential energy is $u(r) = q_1q_2/(4\pi\epsilon_0\epsilon_r r)$ and the force $f(r) = -\mathrm{d}u(r)/\mathrm{d}r$ is $f(r) = q_1q_2/(4\pi\epsilon_0\epsilon_r r^2)$, the justification is high as is seen in Figure 5.3 where the force-distance is a decreasing straight line on the log-log graph, q_1 and q_2 are protons, and r the distance in metre.

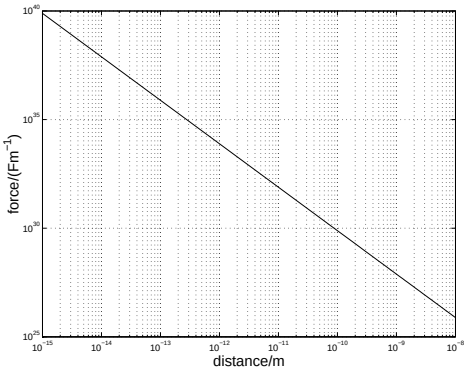


Figure 5.3 Coulomb force *vs* distance in log-log scale..

The effects of the Coulomb force are prominent when sizes of the particles are small. In quantum mechanics where the scale is atomic, for instance, the problem becomes one in which there are a large number of particles interacting with one another, *i.e.* an n-particle problem. In theory the Hamiltonian operator can be applied and then the problem solved numerically. But in general this is not possible due to the too many particles involved in the calculation. Theoretical solution is possible by the various methods of approximation, for example the Hartree method (*cf* Brown, 1972) where the best wave function is found in terms of the one electron function, *i.e.* orbitals, Φ, or a Hatree-Fock approximation which reduces the number of equations to $n/2$.

§ 5.7 Arbitrarily shaped particles

An arbitrarily shaped particle in a divided space will necessarily have a harder time travelling around compared with a sphere even if, or rather especially when the space is a mathematical one. Small particles are roughly spherical, or so the lore of science says, but when one's technology has led one down to the realm of dimensions in the order of those small particles, for instance the nano technology, then shape does matter a great deal and the universal spherical assumption can no longer suffice.

This is only one of the reasons which justify the investigation into the cases where the ratio of surface area to volume is not a minimum. When physics and motion are involved, this kind of study can incorporate both the continuous trajectory calculation as well as the discrete raster grids of the digital visualisation technology. The best way to hold such data is as matrix maps or masks.

The best and quickest way to define the boundary of an irregularly shaped object while allowing ease of investigation of its interaction with the surrounding is as raster outlines. To do this, I first investigated such outlines computed from the equation of straight line segments which make up the boundary of the object. Hereafter object boundaries are defined without loss of generality as straight line segments that link all the vertices together with no gaps. The first investigation is unsuccessful as it violates this definition by leaving many gaps, as can be seen in Figure 5.4 (a). The second algorithm calculates the grids from the coordinate axis along the direction of which the slope of the edge is minimum. But this too still leaves gaps along the boundaries.

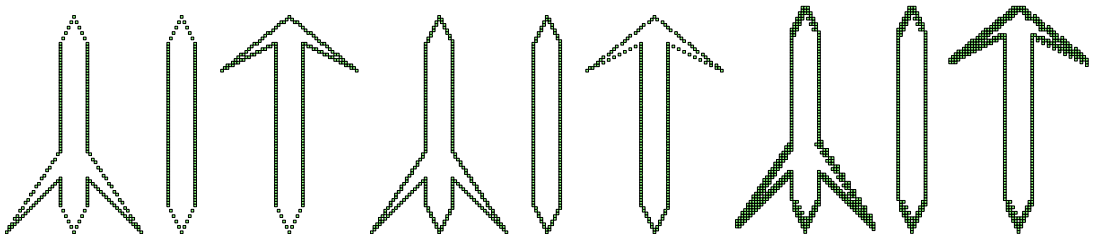


Figure 5.4 *Example of irregular object shapes, Kit in Anglo-Saxon's runic script (cf Freeborn, 1992) created by three different algorithms.*

The object shape is defined by a set of landmark vertices or corners. This must be listed in intended order, otherwise it would be impossible to deduce the order intended unless one assumed that the object is convex.

There are three different programs in § A.22. Only the third one, `tioa3.m` gives a satisfactory result which leave no holes in the surface of the objects. In the second program, `tiao2.m`, `v` is a data set which contains the number of vertices, the vertices, slope, map matrix and the object's dimension. In `tioa3.m`, `v` is somewhat different. It contains the number of vertices, the vertices in real dimension, the grid dimension of the box containing the object, the vertices in grid dimension and the matrix map or image of the object. In order to exploit possible parallelisation, the last program ignores the warnings when there is a division by zero as this means that all the grid positions of the whole line can be operated upon together as a vector quantity.

The third program operates as Algorithm 5.2 does. It utilises the same idea as that used by a child or an artist alike when they draw or paint. Drawing and painting are both one-dimensional process which seeks to produce two-dimensional results. Each pencil- or brush stroke travels along a path the direction of which has one dimension. Unless he uses a very thick brush, there will always be a possibility of gaps forming between stroke lines, which can be annoying because however small they may be one needs to use a great deal of paint in order just to cover them up. This problem can be overcome by painting along two directions, for instance perpendicular to each other especially when the person concerned is a child. At a first glance, or to a novice, this may seem a wasteful practice, especially with dear types of ink. But a little practice and experience will show that this proves in most cases to be more economic than painting along only one direction. And since it can be readily seen that painting in two directions perhaps already uses twice the amount of ink required to cover the paper, it follows that doing so in one direction only would use a great deal more ink than this.

Algorithm 5.2 *Particles or objects with arbitrary shape.*

$(i, j) \leftarrow$ find grid coordinates for (x, y);

find all vectors linking vertices;
$\{p_i\}$ **discretise** these vectors into sets of points;
for all intervals between consecutive p_i's **do**
  draw along y-direction to fill gaps;
  draw along x-direction to fill gaps;
endfor

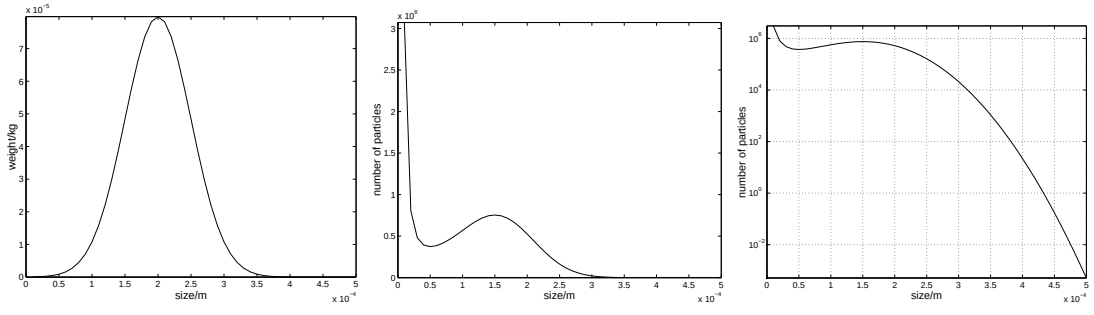
□

The problem of texture quantification is in the difficulty in expressing such quantity by a single parameter. Properties such as coarseness, smoothness, heterogeneity and regularity are finger prints of particles, and are the result of physical and chemical processes. By using fractal geometry, it is possible to describe quantities like texture. A fractal is a set whose metric properties can only be consistently illustrated with a dimension $D > T$, where T is the standard topological dimension. We can express this dimension as $D = T + (1 - H)$, where H is the codimension. Curves can have their roughness described. By giving a fractal number between one and two the space filling ability of the curves is established (*cf* Van Put *et al*, 1994). The first method defines the fractal dimension of a function in terms of the Fourier power spectrum $P(\omega) \approx \omega^{-(2H+1)}$; H and D are determined by doing a linear regression on the log-log plot of the observed power spectrum as a function of frequency. Another method defines fractal dimension in terms of how the variance of interpixel differences changes with distance; D is estimated from a log-log plot, a variogram, of the variance of increments versus increments $\sigma^2(x) \approx x^{2H}$.

Techniques used for measuring sizes of particles include sieving, microscopic analysis, electronic particle counters, laser diffraction analysis, permeability-, sedimentation and elutriation methods. Usually arbitrarily shaped particles are characterised by a variety of methods. For a single particle this usually means transforming its property to the corresponding value of a sphere. A particle may be represented by a sphere which has the same volume, surface area, surface area per unit volume, or the area projected perpendicular to the flow direction. It may be compared with a sphere which has the same settling velocity in the same fluid, or a sphere which will just pass through a square aperture of the same size. Or its area projected on to the ground, when it is resting in the position of maximum stability, may be compared with that of a sphere in the same situation. To be consistent in one's choice is more important than which choice one chooses from. The sphericity of a particle is defined as $\Psi = A_V^s / A_p$, where A_V^s is the surface area of sphere which has the same volume as the particle and A_p surface area of the particle itself (*cf* Coulson *et al*, 1991).

§ 5.8 Non Poisson number distributions of particles

Let us look at an aggregate of spherical particles whose average diameter is 200 ± 50 μm, density $\rho = 3.7$, and the total weight one gramme. Assuming the gravitational constant to be 9.81 kg $\cdot$ ms^{-2}. The overall weight is $W = n \cdot w$, where $w = \rho V g$ is the weight of each particle. The particles being spherical their individual volume is thus $V = (4/3)\pi r^3$ and their distributions by weight and by number are shown in Figure 5.5.



(a) (b) (c)

Figure 5.5 (a) *distributions by weight,* (b) *distribution by number in normal scale and* (c) *with log scale in the y-axis.*

In particle size classification the separation size, D_{50}, is the size where there is an equal chance of particles being reported to the fine or coarse fraction. The grade efficiency curve is normally plotted between the weight fraction to coarse product and the normalised particle size D/D_{50}. The choice of classification equipments depends on many factors, for instance the size of the particles in question

and their electrical or magnetical property. For fine particles, Treasure (1965) discusses several types of classifier, namely the solid-bowl centrifuge, the Hosokawa micro-separator and the Head, Wrightson air classifier all three of which have $D_{50} \propto \sqrt{q}/\omega$, the hydrocyclone where $D_{50} \propto 1/q^{0.5}$ and $q \propto p^{0.5}$, and the Alpine Mikroplex classifier where $D_{50} \propto u/(w\sqrt{V})$ where u and w are respectively the radial and tangential velocities and V is the volume air flow rate, and $D_{50} \propto 1/w$ in practice.

The distribution of bubble size in gas fluidised beds had been represented by various kinds of distribution, for example the log normal, Gamma and χ^2 distributions. Rowe and Yacono (1975) preferred the gamma distribution of the volume, that is $\Gamma(V|m,n) = [1/(n^m\Gamma(m))]\, V^{m-1} \exp(V/n)$, in its normalised form $\Gamma(v) = \left[m^m \exp(-mV)V^{m-1}\right]/\Gamma(m)$.

Sieve testing is affected to blinding materials, that is parts of materials under test which lodge themselves in the apertures of the sieve. To account for this Rose and English (1973) give a general rule of thumb which says that a particle for which $\theta < \tan^{-1}\gamma$ will blind the sieve. Here θ is the angle that the line of reactive force exerted by the material on the particle, which is normal to the contact surface, makes with the horizontal plane. Then they give the empirical value of γ to be approximately $20°$, which results in $\theta < 0.3358$ radians as being the criterion for blinding to occur. For a spherical particle this gives the ratio $x/r = \cos\theta > 0.9441$, where r is the particle's radius and x half the minimum distance between two normal reactions which act on the particle. Therefore the radius of particles which blind the sieve must be such that $r < 1.0592x$, which corresponds to the aperture of 1.0592 times greater than the aperture if we assume that the aperture is $2x$ through out, or equivalently that the openning has its corresponding faces vertical.

It would be of a theoretical interest to prove this equation $\theta < \tan^{-1}\gamma$ given by them. Also, the angle of friction, γ, is not a constant but varies for different pairs of materials in contact with each other, as well as changes the value when we change the medium surrounding them to a different type, or even when the moisture content changes for that matter. All of this goes to say that the value $1.1d_{\min}$ extensively used in chemical engineering literature is probably nothing more than an engineering rule of thumb based on two conjectures, one in the limiting value of θ, the other in the value of γ. But it gives a simple and convenient criterion for use when doing computer simulations, which are nothing but means to visualise the conceptualised physics anyway. Other values of $\theta_{\max}$ should give the same qualitative result. We may, for instance, choose the coefficient of our $d_{\max}$ to be 1.02, corresponding to $\theta > 1/\cos 10°$, instead of 1.06 which corresponds to the $\theta > 1/\cos 20°$ used above.

Since no explanation of the formula regarding θ given by Rose and English has been given elsewhere, I arrive at my own derivation as follows. First, let us recall from our sixth form physics how friction can be described by either the coefficient of friction or the angle of friction, both of which are merely different sides of the same coin. The angle of friction is also known as the angle of frictional resistance, the internal angle of friction and the angle of shearing resistance. For clean sand it is approximately the angle of repose. It reduces with moisture content and is zero for saturated clay. For solid it is approximately the angle of inclination of the surface of one material at the point when a block of another material, placed on top of the former, starts to slide. This is precisely the method by which the coefficient of friction is determined. There are two different coefficients of friction for each pair of material, namely the static- and the kinetic coefficients of friction, the former having a slightly higher value than the latter, and both are defined as being the ratio between the limiting friction and the normal reaction, $\mu = f_k/F_N$. This gives rise to the formula for the friction being $f_k = \mu F_N$, which is equal to $mg\sin\theta$ at the point when the object of the second material slips, θ being the inclination of the plane made by the first material. Therefore we have $\mu = \tan\theta$, where μ is the coefficient-, and θ the angle of friction.

Coming back to our formula in question, $\theta < \tan^{-1}\gamma$. Figure 5.6 shows a spherical object sitting on top of an aperture in a sieve. From Figure 5.6, $\alpha + \theta = \pi/2$. Without loss of generality, assume $\gamma < \pi/4$. Let $\mu = \tan\alpha_c$ be the coefficient of friction at some critical value α_c. Furthermore assume that α_c is small, so that we can approximate μ by the angle of friction γ. Then we have $\alpha_c = \tan^{-1}\mu = \tan^{-1}\gamma$. It follows that the particle will block the pore whenever $\alpha < \alpha_c$, whereas it will pass through the latter when $\alpha = \pi/4$.

Because we know from experiments that some of the particles does lodge against and blind the pore, there must exist some $\alpha_c < \beta < \pi/4$ such that this occurs. For the reason that the number of particles that blind is empirically small we know that β is small. Since we already have the relationship among α, θ and $\pi/4$, namely $\alpha+\theta = \pi/4$, and since α_c is also small from our assumption, then in order not to unnecessarily introduce another parameter into what necessarily already contains some error due to all the approximations so far made, we let $\beta = \alpha_c$. Now we may say that the particles will blind if and only if $90° - \alpha_c = 90° - \beta < \alpha < 90°$. For other remaining values of α, *viz.* both $\alpha_c \leq \alpha \leq 90°$ and $\alpha < \alpha_c$, in other words $0 \leq \alpha \leq 90° - \alpha_c$, then, particles must block. But $\theta = 90° - \alpha$, therefore the particles pass through when $\theta = 0$, blind when $0 < \theta < \alpha_c$ and block when $\alpha_c \leq \theta \leq 90°$. So the blinding particles have $\theta < \alpha_c = \tan^{-1}\gamma$ and the approximation is explained though not *q.e.d.*'ed.

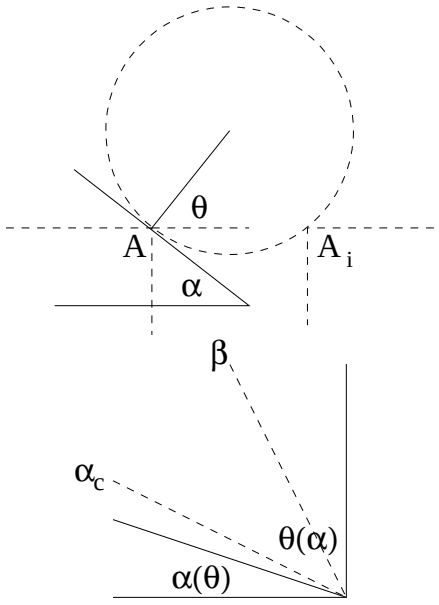


Figure 5.6 Spherical object on a sieve.

Let A_a be the available area free from blinding material at some instant, A the physical area of the sieve cloth, G the area of cloth blinded by a unit mass of blinding material, and w_0 and w the mass of blinding material on the sieve respectively at $t = 0$ and at some $t > 0$. Then $A_a = A - G(w_0 - w)$ and $\mathrm{d}w/\mathrm{d}t = -k_1 w A_a$, where k_1 is the diffusion coefficient, and then $\mathrm{d}w/[w(b+w)] = -a\mathrm{d}t$, where $a = -k_1 G$ and $b = (A/G) - w_0$. Assume $b \neq 0$. At $t = 0$, $w = w_0$. It follows that $w = [bw_0 \exp(-abt)] / [b + w_0(1 - \exp(-abt))]$.

Furthermore, let W be the total weight of material on the sieve, W_∞ the residue on the sieve for a theoretically infinite time of sieving, and K a constant. Then, $\mathrm{d}W/\mathrm{d}t = -K(W - W_\infty)A_a$, and it follows that $\mathrm{d}W/(W - W_\infty) = -KG(b+w)\mathrm{d}t = -KGb\mathrm{d}t - KGb\exp(-abt)\mathrm{d}t/(q - \exp(-abt))$, where $Q = (b/w_0) + 1$. Again assume $b \neq 0$, and $W = W_0$ at $t = 0$. Then, $\ln[(W_0 - W_\infty)(W - W_0)] = K\{Gbt + (G/a)\ln[(((b/w_0) + 1) - \exp(-abt))/(b/w_0)]\}$, and then $W = W_\infty + (W_0 - W_\infty)\delta$, where $\lambda = b/w_0 = (A/Gw_0) - 1$ and $\delta = [\exp(a\lambda w_0 t)(\lambda + 1)/\lambda - 1/lmd]^{-KG/a}$. When the total amount of blinding material is sufficient to bind the sieve completely, $A/Gw_0 < 1$ and so $-1 \leq \lambda < 0$, when it is exactly enough to completely glind the sieve, $\lambda = 0$, otherwise $A/Gw_0 > 1$ and $0 < \lambda < \infty$, Gw_0 being the area of sieve cloth which will be blinded by all the blinding material present. At $t \to \infty$, $\delta = 0$ when $0\lambda < \infty$ and $\delta = (-\lambda)^{KG/a}$, a positive fraction, when $-1 \leq \lambda \leq 0$.

6

§ 6. Filtering membranes

Tanemura *et al* (1983) give a good algorithm which is both powerful and simple for constructing the three-dimensional Voronoi tessellation. The circumcentre of a DT being a vertex of the VT and the circumsphere of a DT being empty, this second condition of which is known as the *contiguity condition*, they arrive at an algorithm which works for both nondegenerate and degenerate cases. Algorithm 6.1 summarises their algorithm. Associated with the atom i, let V_i be the polyhedron, v_i vertex atoms of V_i, and S_i set of atoms surrounding i, $S_i \supset v_i$. Here $H_i(\alpha\beta|\gamma)$ is the half-space determined by $\{i, i_\alpha, i_\beta\}$ and does not contain i_γ, $S_i(\alpha\beta|\gamma) \subset S_i \subset H_i$.

Algorithm 6.1 *Voronoi construction in three dimensions, Tanemura et al (1983)*

find i_1, nearest, and thus contiguous, to i;
$C_i \leftarrow i_1$;
find $j = i_2$ s.t. $\{i, i_1, i_2\}$ has the minimum circumradius among all $\{i, i_1, j\}$, $j \in S_i$, $j \neq i_1, i_2$;
$C_i \leftarrow i_2 \cup C_i$;
find $j = i_3$ s.t. $\{i, i_1, i_2, i_3\}$ has the minimum circumradius among all $\{i, i_1, i_2, j\}$, $j \in S_i$, $j \neq i_1, i_2$;

$V_i \leftarrow \{i, i_1, i_2, i_3\}$;
$C_i \leftarrow i_3 \cup C_i$;
clear $\{m_\alpha\}$ **and** $\{l_{\alpha\beta}\}$;
$\alpha \leftarrow 1$;
for all $i_\alpha \in C_i$ **do**
if $m_\alpha = 1$ **then**
$\alpha \leftarrow \alpha + 1$;
else
find $\{i, i_\alpha, i_\beta, i_\gamma\} \in T_i$, which has $\{i, i_\alpha\}$ in common and where $l_{\alpha\beta} = 1$;
$S_i(\alpha\beta|\gamma) \leftarrow S_i \cap H_i(\alpha\beta|\gamma)$;
describe a circumsphere to each quartette $\{i, i_\alpha, i_\beta, j\}$, where $j \in S_i(\alpha\beta|\gamma)$ and $m_j \neq 1$;
find $j_{\min}$ for which the centre of the circumsphere of $\{i, i_\alpha, i_\beta, j_{\min}\}$ has the minimum signed z-coordinate value among all circumspheres obtained, z-axis being normal to $\{i, i_\alpha, i\beta\}$ and lies towards the side of $H_i(\alpha\beta|\gamma)$;
$T_i \leftarrow \{i, i_\alpha, i\beta, i_\delta\} \cup T_i$;
if $i_\delta \notin C_i$ **do**
$C_i \leftarrow i_\delta \cup C$; $l_{\alpha\beta} + +$; $l_{\alpha\delta} + +$; $l_{\beta\delta} + +$; $l_{\beta\alpha} + +$; $l_{\delta\alpha} + +$; $l_{\delta\beta} + +$;
endif
if all $\{l_\alpha\}$ are equal to 2, $m_\alpha \leftarrow 1$;
if all $\{l_\beta\}$ and/or $\{l_\delta\}$ are equal to 2, $m_\beta \leftarrow 1$ and/or $m_\delta \leftarrow 1$;
endif
endfor
find geometrical quantities of Π_i;) □

Algorithm 6.1 is justified by proofs of theorems which confirm that atoms nearest to each other are contiguous to each other, a triangle with the minimum circumradius is a face of a DT, which means that all its three atoms are also mutually contiguous, likewise a tetrahedron with the minimum circumradius is a DT, all its four atoms are mutually contiguous, two DT's sharing a plane have their fourth vertices at a signed minimum distance on either side of the plane they share. Some interesting geometrical lemmas help towards the proof of these theorems.

Jackson (1994) studies porous media and represent them using Voronoi models. His interests are in the synthetic membranes, which can be symmetric or asymmetric, homogeneous or heterogeneous structure, neutral or charged, and passive or active transport. The resistance to mass transfer is mainly in the top layer if it exists. This is a dense layer 0.1–0.5 μm thick which lies on top of a porous sublayer 50–150 μm thick. Regular tessellations in two dimensions come in 11 distinct tilings called Archimedean tilings. Tessellations by random lines in 2-d have the equation of the lines $x\cos\theta + y\sin\theta - h = 0$, where $0 \leq \theta \leq \pi$ and $-\infty < h < \infty$, which is analogous to tessellations of random planes in 3-d with the equation of a plane $x\sin\theta\cos\phi + y\sin\theta\sin\phi + z\cos\theta = 0$, where $0 \leq \phi \leq \pi$. The definition he gives of the *Voronoi polygon* is $\bigcap_{i\neq j}^{N} H(x_i, x_j)$, where the half-plane $H(x_i, x_j)$ is $\left\{x \in E^2 | d(x, x_i) \leq d(x, x_j); i \neq j\right\}$. The bisector is $B(x_i, x_j) = \left\{x \in E^2 | d(x, x_i) = d(x, x_j)\right\}$. There are $n_c + n_b - 2$ triangles in the corresponding Delaunay triangulation if in the Voronoi tessellation there are n_c interior- and n_b boundary polygons. Scanning electron microscopy is explained and several SEM pictures of cellulose nitrate are given. AVS is used to do the pore perimeter calculation by connecting the modules `read any image` with `image measure` and *image viewer*, the latter two of which are in turn connected together. To calculate the pore area connect `read any image` with `sketch`, `histogram`, and `image viewer`. Then connect `sketch` with `histogram`, `sketch` with `image viewer`, and `histogram` with `print field`. A 3-d process is a homogeneous Poisson point process if the number of points in any region of volume V has a Poisson distribution with parameter λV and the random variables corresponding to the number of points in disjoint regions are independent among one another. Among the characteristics calculated are, in our terminology, the perimeter of polyhedron $P_c = \sum_{i=1}^{n_c^e} e_c^i$, the surface area $A_c = \sum_{i=1}^{n_c^f} \sum_{j=1}^{n_c^{t,i}} A_t^{c,i,j}$, and the volume is $V_c = \sum_{i=1}^{n_c^f} \sum_{j=1}^{n_t^{c,i}} V_t^{c,i,j}$. His numerical results given (see also, Jackson *et al*, 1999) are $n_f^e = 5.203390$, $n_c^e = 39.191489$, $n_c^v = 26.127660$, $n_c^f = 15.063830$, $P_c = 17.358196b$, $A_c = 5.910997b^2$, b^3 being the mean cell volume; the coordination number is of course 4. The distribution of n_c^v shows the distinct Voronoi characteristic of containing exclusively of even numbers, whereas that of n_c^e the equally distinct characteristic of being exclusively in multiples of three. This has been explained as odd number implying an impossible fractional number of edges, there being 3/2 edges as there are vertices. For the cross section data he found $n_c^e = 5.831$ and $P_c = 3.563s$, s^2 being the mean polygon

area. In membranes, if the diameter of a particle is less than that of a pore then the particle will entrain the pore unless it should encounter other fouling particles, but if the particle is bigger than the pore then it will be retained. He assumes fouling particles to be spherical. Ascribed to each face is the diameter or size of its largest inscribed circle. Algorithm 6.2 is used for this purpose; d is the largest inscribed circle diameter sought for.

Algorithm 6.2 *Find the largest inscribed circle*

$d \leftarrow 0$
for *all combinations of three edges of the face*
  find the corresponding *triangle*
  find the *largest inscribed circle* of this triangle
  if the *centroid* lies within the face **then**
    if the *perimeter* lies within the face **then**
      if the *diameter* $> d$ **then**
        $d \leftarrow$ the *diameter*
      endif
    endif
  endif
endfor □

In carrying out his simulation, the mean particle diameter $\bar{d}_p$ is calculated from the mean pore diameter $\bar{d}_v$ by $\bar{d}_p = k\bar{d}_v$ where k is a constant. The corresponding standard deviation of particle sizes is $\sigma_p = \sigma_v(\bar{d}_p/\bar{d}_v)$. The distribution of particle diameters is dependent on both k and another dimensionless parameter $m = (\sigma_p/\bar{d}_p)/(\sigma_v/\bar{d}_v)$. Then the central limit theorem gives the frequency distribution and thus the diameter of the i^{th} particle as $d_i = \bar{d}_p + \sigma_p \left(\left(\sum_{j=1}^{q} r_j - q/2 \right) / \sqrt{q/12} \right)$, where $0 \leq r_j \leq 1$ are random variables and $q \geq 12$. Interaction between particles and membranes starts from Algorithm 6.3 which has been adapted from Jackson (*ibid.*).

Algorithm 6.3 *Interaction between particles and surface pores.*

while another *particle* exists **do**
  $i \leftarrow i + 1$;
  $p_i \leftarrow$ *particle* ;
  find particle entry *coordinates*;
  $c_i \leftarrow$ of the nearest *Voronoi cell*;
  $f_i \leftarrow$ the nearest *face* **of** c_i;
  if the face is *fouled* **then**
    for all the *neighbouring faces* f_j
      if f_j is *not fouled* **then**
        $\{f\} \leftarrow f_j$;
      endif
    endfor
  else
    $\{f\} \leftarrow f_i$;
  endif
  if $\{f\}$ is not empty **then**
    $j \leftarrow 0$
    while particle *travels* and $j < |\{f\}|$ **do**
      $j \leftarrow j + 1$;
      if $d_p > d_i$ **then**
        if the centre of p_i lies outside the largest circumscribed circle of f_i **then**
          $f_i \leftarrow$ the neigbouring face f_i of the adjacent cell;
        endif
        if $d_p > 1.1 d_i$
          particle *cakes* or *blocks* surface;
        else
          particle *blinds* surface face;
        endif
      else
        particle *enters* membrane;

```
        endif
      endwhile
    else
      particle cakes or blocks surface;
    endif
  endwhile                                                    □
```

Backflushing the membrane may clean blocking- but not blinding particles. Once inside the membrane, each particle independently and randomly walks until it can get out of the membrane or can go no further and thereby necessarily fouls a cell. The interpretation shown here as Algorithm 6.4 is an adaptation of that given by Jackson (1994).

Algorithm 6.4 *Interaction between each internal particle and the pores.*

```
for each internal particle p_i
  {c} pores to enter;
  j ← 1
  while j ≤ |{c}| and p_i has not entered a polyhedron do
    find pore to enter;
    if the polyhedron is fouled then
      particle enters the polyhedron;
      if there is no space in the polyhedron then
        particle blocks or blinds pore;
      endif
    endif
    i ← i + 1
  endwhile
endfor                                                        □
```

On the cake, the particles drop, roll, or nest among one another. The cake formation is described as Algorithm 6.5 which closely follows the description which he gave.

Algorithm 6.5 *Particle cake formation by Jackson (1994)*

```
while next particle exists do
  p_i ← next particle;
  if route not directly leads to surface face then
    p_i and p_1 collide;
    find rolling path;
    if p_i does not roll off then
      p_i and p_2 collide;
      if p_i does not roll off then
        p_i and p_3 collide;
        p_i comes to rest;
      endif
    endif
  endif
endwhile                                                      □
```

Fifty particles is the empirical rule of thumb for the number of neighbouring particles to consider at the surface and cake level. Good care has been taken by his program to ensure that no two particles within this set overlap. All of his particles are gentle for they bounce not but often roll, for these no hard billiard balls but mathematical particles with assumptions. In this way a spherical particle would first vertically drops, then touches another particle and starts rolling off the latter along its surface. Once the two particles are abreast with each other they part. A dropping particle can either roll off one- or simultaneously two particles at a time. Rolling simultaneously off three particles is not physically possible. For if we picture i meets 1, rolls off 1, meets 2, starts to roll simultaneously off 1 and 2, and then meet 3. At this point it can either choose to give up 1 to roll simultaneously off only 2 and 3 or, it nests on the cradle of 1, 2 and 3, and thereby stops. There are three phases or layers associated with this membrane packing model. These are the cake-, surface- and internal layers. For the internal layer the thickness is limited by the membrane thickness, for the surface layer by the maximum diameter of the dropping particles, and for the cake layer by their amount and sizes. The flux decline across the membrane model is calculated by Algorithm 6.6.

Here C_n is the particle concentration, P_0 initial pressure, P_L final pressure, J_m initial volumetric flux rate, that is to say, clean solvent flux for membrane, d diameter of the circular membrane, n the number of particles to be dropped on to the membrane in this simulation, L the depth of the packing retion, $\bar{d}_p$ the mean spherical particle diameter, μ the solvent viscosity, ϵ_v the free volume, that is to say, the ratio between the volume of voids and volume of bed, v initial velocity of the fluid, P_s pressure between the cake and the surface fouling layer, R_1 the pressure-flux relationship for the particles in contact with the membrane, R the pressure-flux relationship of the membrane,

Algorithm 6.6 *Flux decline across the membrane model.*

define Δt, C_n, P_0;
$v \leftarrow (4/\pi)J_m/d^2$;
$t \leftarrow 0$;
while simulation should last **do**
  $t \leftarrow t + \Delta t$;
  $n \leftarrow \mathrm{f}(\Delta t, t, J_m)$;
  find the *packing density, dimensions* and *number of particles* of the cake;
  find the *packing density, dimensions* and *number of blocking-* and *blinding particles* of the surface fouling layer;
  find the *packing density, dimensions* and *number of blocking-* and *blinding particles* of the internal fouling layer;
  $P_c \leftarrow$ the *pressure drop across the cake* **calculated from** $v = \frac{(P_0-P_L)}{L}\frac{\bar{d}_p^2}{150\mu}\frac{\epsilon_v^3}{(1-\epsilon_v)^2}$
  $P_s = P_0 - P_c$
  $P_l \leftarrow$ the *pressure drop* for the *surface layer*;
  $P_i \leftarrow$ the *pressure drop* for the *internal fouling particles*;
  $R_1 \leftarrow \frac{P_l+P_i}{J_m}$;
  $R_2 \leftarrow R_1 + R$;
  $J_m \leftarrow P_s/R_2$;
endwhile □

Flux is the rate of flow by weight. The number of particles per second is obtained by multiplying† the concentration by the volumetric rate. And if we divide this by the area of the membrane we get the number of particles per second per unit area. The random walk of particles during removal by backflushing is shown here as Algorithm 6.7.

Algorithm 6.7 *The random walk of particles during removal by backflushing*

remove cake and all surface blocking particles;
while *next particle exists* and *backflushing* **do**
  backflushing $\leftarrow$ false;
  if not blocking **then**
    remain blinding;
    if there are more particles in cell **then**
      blocking $\leftarrow$ true;
    else
      no more particle;
      backflushing $\leftarrow$ true;
    endif
  else
    blocking $\leftarrow$ true;
  endif
  if *blocking* **then**
    do
      if cell inlet pore unfouled or fouled and the particle passes through adjacent pore **then**
        move into the next adjacent cell;
      else
        remain blocking;
        backflushing $\leftarrow$ true;
      endif

† It is possibly a typing error in Jackson (1994) when he says *dividing* instead of *multiplying* here.

while the particle is still within the membrane **enddo**
particle has left membrane;
backflushing $\leftarrow$ true;
endif
endwhile □

Filters are sometimes made of non-woven materials, in which case they are modelled as tessellation by random straight lines. Wilkinson *et al* (1986) studies this type of filter by modelling it as intersecting random rods. The position and orientation of a rod is defined by one point, an angle $0 \leq \theta_i$ 180° and the diameter of the fibre d_i. The free area is $\varepsilon_A = (A - \sum A_s)/A$, where the total fibre cross sectional area is $\sum A_s = \sum_m l_i d_i - \sum_m \sum_n \delta_{ij} d_i d_j \operatorname{cosec} \theta_{ij}$, where l_i is the length of the fibre, δ_{ij} is 1 if lines i and j intersects and 0 otherwise, θ_{ij} the angle of intersection of the fibres. Their model worked well for small fibre diameters compared to those of the particles.

Algorithm 6.8 rewrites their method in an algorithmic form. Here $0 < r_j < 1$ is a random number with uniform distribution, d_c the critical diameter, that of a circle which just fits the inside an irregular polygon.

Algorithm 6.8 *Non-woven fibre simulation, Wilkinson et al, 1986.*

for all particles i **do**
$d_i \leftarrow \bar{d} + \sigma(\sum_n r_k - n/2)/\sqrt{R/12}$;
choose the particle position $0 \leq x_i \leq X$ and $0 \leq y_i \leq Y$;
for each layer of the filter j **do**
$d_j^c \leftarrow$ the largest circle which fits inside the polygon;
if $d_k > d_j^c$ **then**
the particle is retained at this position;
endif
endfor
endfor

□

The fibre here is in the form of fibre matt, which stacks one upon another in layers. The equation for the volume ratio suspension to inlet of a fibre matt, $V_n/V_0 = \exp(-\xi n)$ where ξ is the mean capture efficiency of the filter, is similar to that of the concentration ratio in deep bed filtration, $C/C_0 = \exp(-Kx)$. The mean capture efficiency is related to the layer efficiencies by $\xi = 1 - [(1-\xi_1)(1-\xi_2)\cdots(1-\xi_n)]^{1/n}$, and is dependent on $\bar{d}$, σ, ε_A, $\bar{d}_f$, σ_f, where the subscript f means *filament.*

Jafferali (1995) studies Voronoi tessellation and applies it to microfiltration. The program which he used to create the Voronoi structure has the algorithm of Algorithm 6.9.

Algorithm 6.9 *Construction of 3-d Voronoi network (Jafferali, 1995)*

find set S which contains m nuclei nearest to the nucleus point p;
find the nucleus q which is nearest to p;
find a nucleus $r \in S, r \neq q$, such that the triangle pqr has the minimum circumradius, that is one facet of the Delaunay triangulation;
do
find $s \in S, s \neq r \neq q$ such that the circumradius of the tetrahedron $pqrs$ is minimised, that is a Delaunay tetrahedron
find $m, n, o \in S$ such that $m \neq n \neq o \neq q \neq r \neq s$ and the tetrahedra $pqrm$j, $pqsn$ and $prso$ all have minimum circumradii, these are also Delaunay tetrahedra
until each triplet pp_ip_j is in two Delaunay tetrahedra, that is every face of every Delaunay tetrahedron containing p as a vertex is closed **enddo**

To find the centre of the circumscribed circle of the triangle ABC, let Π_A, Π_B and Π_C be the planes perpendicular respectively to the sides opposite to A, B and C. Respectively let n_A, n_B and n_C be the vectors normal to Π_A, Π_B and Π_C, and a, b and c the position vectors of A, B and C. Then Π_A, Π_B and Π_C will intersect at a point the position vector of whom is r which is obtained by solving the three simultaneous equations $(r - n_i) \cdot a = 0$ where $i = A$, B, C, $n_B = n_C \times n_A$, $n_A = b - c$, $n_C = b - a$, $d = 0.5(a + b)$, $e = 0.5(b + c)$. It follows that $r_x = [m_1 - (r_x a_y + r_z a_z)]/a_x$, $r_y = m_6 + m_7 r_z$ and $r_z = (m_4 - m_6)/(m_7 - m_5)$ where $m_1 = r \cdot a = a \cdot n_B$, $m_2 = r \cdot d = d \cdot n_C$, $m_3 = r \cdot e = e \cdot n_A, m_4 = (m_2 a_x - m_1 d_x)/(d_y a_x - a_y d_x)$, $m_5 = (a_z d_x - d_z a_x)/(d_y a_x - a_y d_x)$, $m_6 = (m_3 a_x - m_1 e_x)/(e_y a_x - a_y e_x)$, and $m_7 = (a_z e_x - e_z a_x)/(e_y a_x - a_y e_x)$.. The centre p of the

circumscribed sphere is obtained by solving the four simultaneous equations the first one of which is $(p_x - A_x)^2 + (p_y - A_y)^2 + (p_z - A_z)^2 = r^2$ while the other three have B, C and D in place of A. Solving these we get $p_x = (m_2 - m_4)/(m_3 - m_1)$, $p_y = m_1 x + m_2$, $p_z = [2p_x(c_x - d_x) + 2p_y(c_y - d_y) + e_2 - e_1]/[2(d_z - c_z)]$ where $m_1 = [(a_x - b_x)(d_z - c_z) + (d_x - c_x)(b_z - a_z)]/[(c_y - d_y)(b_z - a_z) + (b_y - a_y)(d_z - c_z)]$, $m_2 = [(e_4 - e_3)(b_z - a_z) - (e_1 - e_2)(d_z - c_z)]/[2(c_y - d_y)(b_z - a_z) + 2(b_y - a_y)(d_z - c_z)]$, $m_3 = [(a_x - d_x)(b_z - c_z) + (b_x - c_x)(d_z - a_z)]/[(c_y - b_y)(d_z - a_z) + 2(d_y - a_y)(b_z - c_z)]$, $m_4 = [(e_2 - e_3)(d_z - a_z) - (e_1 - e_4)(b_z - c_z)]/[2(c_y - b_y)(d_z - a_z) + 2(d_y - a_y)(b_z - c_z)]$, $e_1 = a_x^2 + a_y^2 + a_z^2$, $e_2 = b_x^2 + b_y^2 + b_z^2$, $e_3 = c_x^2 + c_y^2 + c_z^2$, and $e_4 = d_x^2 + d_y^2 + d_z^2$. Statistics obtained from his simulation include n_c, n_c^f, n_c^v, n_c^e, perimeter of polyhedron, area of face, and volume of polyhedron. The number of faces and the number of edges are found during face construction the method of which is Algorithm 6.10.

Algorithm 6.10 *Find faces of a polyhedron, Jafferali (1995).*

find the set of all the n vertices of the polyhedron, $\{v\}$;
for all combinations of three vertices v_i, v_j and v_k of $\{v\}$ **do**
  find the plane Π which contains v_i, v_j and v_k;
  if all the remaining vertices lie on one side of Π **then**
    v_i, v_j, v_k **together with all the other vertices which lie on** Π **form a face;**
  endif
enddo □

To find out whether all the vertices lie on one side relative to a plane, find whether the sign of the distance from each of these points to the plane changes. This distance is $d = \overrightarrow{pa} \cdot n = (a - p) \cdot n$ where n is the unit normal vector, $n = \overrightarrow{pa} \times \overrightarrow{bc}/|\overrightarrow{ba} \times \overrightarrow{bc}| = [(a - b) \times (c - b)]/|(a - b) \times (c - b)|$, and a, b and c form a plane while p is the point in question. To arrange the vertices of a face in cyclic order, first pick one point among them and then find the angle made by the lines from it to every two combination of the remaining vertices, by $\theta = \cos^{-1}[(\overrightarrow{v_p v_i} \cdot \overrightarrow{v_p v_j})/|\overrightarrow{v_p v_i}| \cdot |\overrightarrow{v_p v_j}|]$. Two such lines which maximises the angle are both edges, and the angle from either one of them to each of the remaining vertices increases as we tread the edges from one vertex to another in succession.

Nuclei points of the modified point process where $d \geq d_{\min}$ are generated by Algorithm 6.11.

Algorithm 6.11 *Nuclei points of the modified point process, Jafferali (1995).*

n; $d_{\min}$;
find p_1;
$i \leftarrow 1$;
while $i < n$ **do**
  $i \leftarrow i = 1$;
  find p_i;
  for $j = 2$ to $(i - 1)$ **do**
    find $d(p_i, p_j)$;
    if $d(p_i, p_j) < d_{\min}$ **then**
      $i \leftarrow i - 1$;
    endif
  endfor
endwhile □

The statistics that he found for these modidied Voronoi structures are shown here again in Table 6.1.

$d_{\min}$	n_c	$\min(n_f)$	$\max(n_f)$	$\min(n_v)$	$\max(n_v)$
0.0	226	8	23	12	42
0.2	218	8	24	12	44
0.4	215	8	23	12	44
0.6	226	8	24	12	44
0.8	229	9	21	14	38
1.0	214	10	20	16	36
1.2	204	10	20	16	36

Table 6.1 *Statistics of the modified Voronoi cells, Jafferali (1995).*

His asymmetric Voronoi tessellations have cell volume increasing with the z-coordinate position of the cell. The procedure for finding the $d_{\min}$ between two cells in such structure is described as

Algorithm 6.12. Here the volume ratio is $\delta = V_0/V_1$, and thus $\delta = y_0/y + 1$, $d_{\min} = mz_i + c$ where $m = (y - y_1)/(z - z_1)$.

Algorithm 6.12 *Asymmetric Voronoi tessellation, Jafferali (1995).*

δ;
$y_1 \leftarrow 1$;
$y_0 \leftarrow \delta$;
$c \leftarrow \delta^{1/3}$;
$m \leftarrow (1 - c)$;
$d_{\min} \leftarrow k[(1 - c)z + c]$; □

The statistics that he found are given for reference as Tables 6.2 and ℵ. Real numbers have been rounded to leave at most four decimal points to make it easier to read (*cf* Jafferali, 1995).

	point distribution		
	Poisson	**modified**	**asymmetric**
n_c	1276	1171	1069
$\sum_i n_i^v$	33340	29946	27749
n_c^v	26.1285 ± 6.4207	25.5730 ± 3.5573	25.9579 ± 3.5301
$\sum_i n_i^e$	50072	44945	41643
n_c^e	39.2414 ± 9.6235	38.3817 ± 5.3317	38.9551 ± 5.2740
$\sum_i n_i^f$	19284	17341	16032
n_c^f	15.1129 ± 3.2114	14.8087 ± 1.7821	14.9972 ± 1.7514
$\sum_i V_i$	$0.6073b^3$	$0.6582b^3$	$0.5912b^3$
V_c	4.76×10^{-4}b3$\pm 2 \times 10^{-4}$	5.62×10^{-4}b3$\pm 6.8 \times 10^{-4}$	5.53×10^{-4}b3$\pm 6.0 \times 10^{-4}$
$\sum_i p_i$	$641.3870b$	$664.1398b$	$554.01b$
p_c	1.0053b$\pm$0.2290	1.1343b$\pm$0.1058	1.0365b$\pm$0.3255
$\sum_i A_i$	$40.9930b^2$	$44.2189b^2$	$36.4954b^2$
A_c	$0.0321b^2$	$0.0378b^2$	$0.0341b^2$

Table 6.2 *Comparative Voronoi statistics, Jafferali (1995).*

	distribution					
	Poisson		**modified**		**asymmetrical**	
	min	max	min	max	min	max
V	$8.8 \times 10^{-5}b^3$	$1.4 \times 10^{-3}b^3$	$3.9 \times 10^{-4}b^3$	$8.4 \times 10^{-4}b^3$	$9.9 \times 10^{-5}b^3$	$3.7 \times 10^{-3}b^3$
n_c^f	7	28	10	21	10	21
n_c^e	15	78	24	57	24	57
n_c^v	10	48	12	34	12	34

Table 6.3 *More comparative Voronoi statistics, Jafferali (1995).*

The Voronoi domain created has ragged boundaries due to polyhedra protruding. The method used in his thesis is to slice the domain by a horizontal plane and then redefine those vertices and edges on that plane and at the same time reject everything above it. This is more similar to a carpenter filing away at a block of wood than a stone mason choosing his stones. The volume of voids was updated by using AVS which does this by counting the number of pixels contained within a given area. Statistics of an asymmetric Voronoi structure were given. His study concerns with the simulation of dead end filtration and cake formation. The particle size distribution is quantified in terms of the ratio of the mean particle diameter to the mean inlet pore diameter, α, and that of the standard deviation of the particle diameters to the mean particle diameter, β. The first parameter indicates the relative size between the particles and pores while the second one that amongst particles. Thus $\alpha > 1$ would mean that the particles are bigger than the inlet pores, while $\beta = 0$ that particles are mono dispersed. The interaction between the particles and cake is essentially the same as that used earlier by Jackson (1994). The evaluation of pore properties described is equivalent to Algorithm 6.13.

Algorithm 6.13 *Pore property evaluation, Jafferali (1995).*

identify edge pores;
identify inlet- and outlet pores;
for all pores **do**
  find the largest inscribed sphere;
  for all faces of the pore **do**

find the largest inscribed circle;
endfor
endfor □

Identifying edge pores amounts to first finding neighbouring polyhedra, and then finding for all faces whether each of them belongs to one and only one polyhedron. Finding neighbouring polyhedra amounts to finding for all polyhedra pairs whether each of them possess no less than three common vertices.

Particles pass through a face into a pore if they could, but since they are spherical whereas the pores and faces are polyhedral and polygonal, the largest inscibed circle of the facet is computed. If the particles are irregularly shaped, they may fit through a polygon when challenging from a certain direction but not another. However, the assumption of spherically shaped particles provides most authors on the subject with an approximation both satisfactory and convenient, because then there is no need whatever to rotate them. Jafferali (1995) applies an algorithm equivalent to our Algorithm 6.14 to find the inscribed circle for all faces. Essentially this involves computing an inscribed circle of a triangle arisen from every one of the possible combinations of three sides of the face. Then the largest one of such circles which lies inside the face is the inscribed circle of the face. Here a, b and c are the positional vector of A, B and C respectively; n is a unit vector from A to B, q normal to the plane ABC, and m perpendicular to both; s_i are the sides opposite (x_i, y_i). From 3-d each face is transformed into 2-d in such a manner that $A = (x_1, y_1) = (0, 0)$, $B = (x_2, y_2) = (0, y_2)$ and $C = (x_3, y_3)$. From $c - a = x_3 m + y_3 n$, x_3 and y_3 are obtained; z – for *Zentrum* – is the centre (x_z, y_z) of the face in 3-d or its equivalent transformed centre $(\tilde{x}_z, \tilde{x}_z)$; ξ is the nucleus of our polyhedron while ξ_i its neighbouring nuclei. Then the k^{th} combination gives the largest inscribed circle the radius of which is r_k.

Algorithm 6.14 *Inscribed circle of faces, Jafferali (1995).*

for each face **do**
for all the ${}^{p}\mathrm{C}_3$ combinations j of sides of the face **do**
$R \leftarrow 0$;
find A, B and C, the vertices of the circle formed by them;
$n \leftarrow (b-a)/|b-a|$;
$m \leftarrow$ **solve** $m \cdot n = 0$, $m \cdot N = 0$, $|m| = 1$ **and** $|n| = 1$;
$q \leftarrow (b-a) \times (c-a)$;
$\{x_1, y_1, x_2\} \leftarrow 0$;
$y_2 \leftarrow |b-a|$;
$y_3 \leftarrow [(x_3 n_y + y_3 n_y) m_x - (x_3 m_x + n_x y_3) m_y] / (n_y m_x - n_x m_y)$;
$x_3 \leftarrow [(x_3 m_x + n_x y_3) - y_3 n_x] / m_x$;
$p \leftarrow \sum_{i=1}^{3} s_i$;
$S \leftarrow p/2$;
$r_j \leftarrow \left[S \prod_{i=1}^{3} (S - s_i) \right]^{1/2} / S$;
$\tilde{x}_z \leftarrow (s_1 x_3)/p = (s_1 x_3 + s_2 x_1 + s_3 x_2)/p$;
$\tilde{x}_z \leftarrow (s_1 y_3 + s_3 y_2)/p = (s_1 y_3 + s_2 y_1 + s_3 y_2)/p$;
$x_z \leftarrow \tilde{x}_z m_x + \tilde{x}_z n_x$;
$y_z \leftarrow \tilde{x}_z m_y + \tilde{x}_z n_y$;
$z_z \leftarrow \mathrm{ABC}(x_z, y_z)$;
$d \leftarrow |z - \xi|$;
$\zeta \leftarrow 1$;
for all ξ_i **do**
$d_i \leftarrow |z - \xi_i|$;
if $d_i < d$ **then**
$\zeta \leftarrow 0$;
endif
endfor
if $\zeta = 1$ **then**
if $r > R$ **then**
$R \leftarrow r_j$;
$k \leftarrow j$;
endif

endif
endfor
endfor □

Algorithm 6.15 finds the inscribed sphere of each polyhedron. A tetrahedron is formed from each combination of four planes of polygonal faces. The largest of all the inscribed spheres of these tetrahedra is the inscribed sphere of the polyhedron. The intersection of bisecting planes $\cap_{ij=i1}^{ik}\Pi_{ij}$ is the facet in $(3-k \bmod(3))$ dimensions which is equidistant from Π_i and Π_j, $j = 1$ to k. Finding $\Pi = ax + by + cz + d = 0$ amounts to solving $A \cdot (B \times C)$ for a, b, c and d. Finding intersection of planes amounts to solving their equations simultaneously. Here Π_i is the plane containing the i^{th} face, Π_{ij} the plane bisecting Π_i and Π_j.

Algorithm 6.15 *Inscribed sphere of a polyhedron, Jafferali (1995).*

for each polyhedron **do**
$R \leftarrow 0$;
for all the ${}^{n_f}\mathrm{C}_4$ combinations j of the polyhedron **do**
for $i = 1$ to 4 **do**
$A \leftarrow v_i^2 - v_i^1$;
$B \leftarrow v_i^3 - v_i^1$;
$C \leftarrow v_i^4 - v_i^1$;
$\Pi_i(a_i, b_i, c_i, d_i) \leftarrow$ **solve** $A \cdot (B \times C) = 0$;
endfor
find Π_{1i}; $i = 2, 3, 4$;
$p_i \leftarrow \cap_{i=2}^{4}\Pi_{1i}$;
$r_j \leftarrow \mathrm{d}(p_i, \Pi_1)$;
if $r_j > R$ **then**
$R \leftarrow r_j$;
$k \leftarrow j$;
endif
endfor
endfor □

Particles move down vertically in discrete time steps towards the membrane. Upon reaching the latter, the i^{th} particle is greeted by an entrance the aperture of which is circular with radius r_c. If $r_i < r_c$ it penetrates into the pore, if $r_c \leq r_i \leq 1.1r_c$ it blinds and if $1.1r_c < r_i$ blocks the entrance. Blinded particles are not removed by reversing the flow. They become a part of the membrane. Reaching the membrane is by no mean their destiny, and the particles only begin their long and tortuous pelerinage hereafter by embarking on a random walk in the direction towards the centre of the earth. Each particle enters a pore via the face furthest away from the latter, and it leaves via the face closest to it. The position of a facet is that of its inscribed circle. If the lowest facet is not viable, the particle first repositions itself precisely at the centre of the inscribed sphere, and then either moves out from some facet lower than itself or, when all possibilities of travelling having been exhausted, become a residence of that pore. The repositioning part above may seem like a gross approximation, but there is no reasons why this should not make a sound assumption if we consider the fact that a real particle is never spherical in the first place. But if the particle blinds or blocks a facet, then the calculation used by Jafferali (*ibid.*) becomes more accurate. That is $\alpha = \xi + \lambda(z - \xi)$, where $\lambda = |d|/(|d| + |a|)$, $|a| = (r_\alpha^2 + r_z^2)^{1/2}$ and $d = |\xi - z| - |a|$. To summarise, a particle is in a perpetual search for a lowest facet, which of course is closest to the centre of the earth. This kind of study is important because it lets us know which part of the membrane is prone fouling by the particles.

Non-woven fibres are simply stacked layers of material. Extra holes produced by needles which are used to increase their strength are difficult to model and are normaly assumed away before the simulation (*cf* Chan, 1990). Layers are considered as flat when the fibres which make them are considered to be flexible. Capturing of particles must take into account the separation between the layers as well as the usual aperture perpendicular to the flow direction. This is an example of man-made stacked 2-d layers. Examples in nature are numerous, including the honeycomb and cell growth in tissues.

In his simulation, Chan (1990) represents random lines with $(y - y_i) = \tan\theta_i(x - x_i)$, where $0 \leq \theta_i \leq \pi$, a method of interior randomness. Other possible methods include the μ randomness where the line is represented by $x \sin\alpha_i + y \sin\alpha_i = d_i$, where d_i is the distance of the line from a

pole point p_i which is usually taken as the centre of the area, and surface randomness where the position on the boundary of the area is chosen.

Flow channels within a medium follows Poiseuille's equation for flow within capillaries, $Q = \Delta P \pi r^4 / 8\mu\Delta x$, where Δx is the length of the capillary. Darcy's law relates the flow of a fluid through a porous medium to the overall pressure drop, $\Delta P = \mu L u_0 / K$, where u_0 is the superficial fluid velocity, L the depth of the medium and K the permeability coefficient. The permeability coefficient for packed media, $K = \varepsilon^3 / \left[K' S^2 (1-\varepsilon)^2\right]$, where ε is the voidage of the medium, S the specific surface and K' the Kozeny constant. Particles can blind during both the filtration and the backflushing stages. The backflushing efficiency is the ratio between the volume of the particle flushed away and the total volume of particles captured in the media before backflushing.+

§ 6.1 Separation processes

The largest aperture in the screen is the determining factor by which the selection is made, in other words it defines the size of the cut. This is a more logical argument than the one by which it is the mean aperture that defines the cut size (*cf* Rose and English, 1973). This largest aperture determins the smallest size of the particles retained on the screen. But in practice much smaller particles than these will remain on the screen for various reasons the most important one of which is blindings caused by particles with sizes $d_{\min} < d < 1.1 d_{\min}$. The factor 1.1 was first used by Rose and English (*ibid.*), and arises from the angle of friction γ being approximately 20°.

On a sieve at each instant, the area that is free from blinding material is $A = A_s - G(m_0 - m)$ where A_s is the sieve area, G the area blinded by a unit mass of blinding material, m and m_0 the mass of respectively the blinding- and initial blinding material on the sieve. Also, $\mathrm{d}m/\mathrm{d}t = -kmA = -km\left[A_s - G(m_0 - m)\right] = -\alpha m(\beta + m)$, where $\alpha = kG$ and $\beta = [(A/G) - m_0]$.

In surface straining filtration the particles are larger than the pore size, in depth strining filtration both the pore and the particle sizes are commensurate to each other, and in adsortive filtration the particle size is smaller than the pore diameter. Adsortive filters can be made to have higher filtration efficiency, higher capacity adn higher flow rate than depth straining filters (Raistrick, 1986).

Examples of solid-liquid separations normally found in Chemical Engineering are filtration, sedimentation, flocculation, centrifugation, electro-osmotic and electro phoretic dewatering and hydrocyclonic separation. Poole and Doyle (1965) listed some of the aspects in filtration which they thought need further investigation: the effect of rapid pressure increases on the approach to equilibrium porosity in cakes, for instance in rotary filtrations; the relation between drag forces on particles and particle arrangements and shapes; migration of fines within cakes and media; the effect of changing flow paths during washing on porosity.

§ 6.2 Dead-end filtration

Because membranes are porous media, the flux across them follows Darcy's Law, $j = \Delta p/(\mu R)$, where $j = \mathrm{d}V/(A\mathrm{d}t)$, R is the hydraulic resistence, $R = R_m + R_c$ where R_m and R_c are membrane and cake resistances respectively. Assume $R_m = 0$, then $R = R_c = \alpha m/A$, where α is the specific resistance of the cake, m the mass of the filter cake and A the septum area.

§ 6.3 The centre of gravity

The formula for the centre of gravity of objects in general is $c_g = \int xw(x)\mathrm{d}x / \int w(x)\mathrm{d}x$. For a real object the theoretical procedure is complicated, since there is always the possibility that the density is not uniform, but the practical procedure is simple and straight forward, that is by simply hanging the object by strings in various positions and then find the intersection between the lines of string, provided of course that this is possible with the object.

The centre of gravity is important whenever there is a gravitational interaction with an object. Both the translational and rotational motions of objects through space are relative to this point. The c.g. of a triangle is positioned at one third its height whereas that of a half circular disk is $4r/3\pi$ from the straight boundary line. A c.g. always lies on the lines of symmetry when these exist. Other names for c.g. include geocentre, centroid and barycentre.

The centroid of a triangle lies at the intersection of its median. That of a tetrahedron lies at the intersection of all the lines joining the vertices with the centroids of their respective opposite faces. Its coordinates are the mean coordinates of the four vertices.

To find the centroid of a polygon, first tessellate it into triangles and find the centroid of each one of them. Then we have another system of point masses located at the centroids of these triangles, with the mass proportional to their respective areas.

The centroid of a rod is at its mid point, and the weight of a rod is proportional to its length. This can be helpful when we want to find the centroid of a network of rods or tubes.

The centroid of a quadrilateral is the intersection between its two bimedians. This is also the mid point of the lines joining the mid points of the two diagonals.

The medians of a triangle which has its vertices at (a_1, b_1), (a_2, b_2) and (a_3, b_3) are the lines connecting these three vertices to the mid points opposite to them, the first one to (c_1, d_1) where $c_1 = (a_2 + a_3)/2$ and $d_1 = (b_2 + b_3)/2$, and similarly for the other two. Their intersection is obtained by solving their equations, $(d_1 - b_1)/(c_1 - a_1) = (y - b_1)/(x - a_1)$, and so on. This gives the solution as being the average value of the vertices, $[(\sum_3 a_i)/3, (\sum_3 b_i)/3]$.

Having mentioned the centroid, it is natural to add the other two points related to it. On the Euler line also lie the orthocentre, where altitudes intersect, and the circumcentre, where perpendicular bisectors intersect.

Let a_{ij} be $(a_j - a_i)$. Then for the orthocentre we need to solve any two of the three equations of the altitudes. For example, $-a_{23}/b_{23} = (y - b_1)/(x - a_1)$ and $(y - b_2)/(x - a_2) = -a_{13}/b_{13}$ can be simultaneously solved to give $x = [a_1(a_2b_{12} + a_3b_{31}) - (a_2a_3 + b_{21}b_{31})b_{32}] / (a_3b_{21} + a_1b_{32} + a_2b_{13})$ and $y = -[a_{32}(a_1a_2 - a_2a_3 + b_2b_{31}) + a_{31}(a_1a_{23} + b_1b_{23})] / (a_3b_{21} + a_1b_{32} + a_2b_{13})$.

Similarly the circumcentre can be solved from two equations, for instance $(y - b_1)/(x - a_1) = m_1$ and $(y - b_2)/(x - a_2) = m_2$ where $m_1 = -a_{23}/b_{23}$ and $m_2 = -a_{13}/b_{13}$, to give $x = b_{32}(a_2a_3 + b_2b_3)/(a_3b_2 - a_2b_3)$ and $y = a_{32}(a_2a_3 + b_2b_3)/(a_2b_3 - a_3b_2)$.

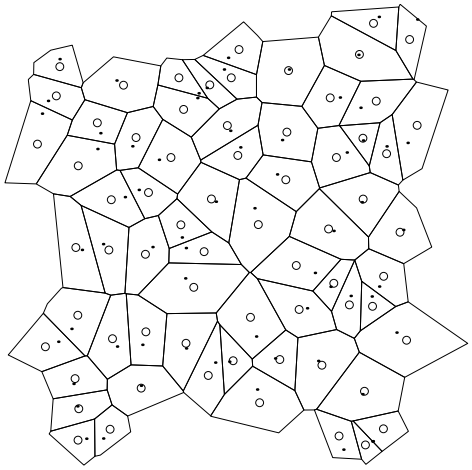


Figure 6.1 shows a 2-d Voronoi tessellation with its nuclei and centroids. Ironically, the centroid represents the position of a cell better than its nuclei, because it positions itself in the most balanced situation relative to the cell, whereas the nuclei does so with regard to its neighbours. The centroids are noticeably more evenly distributed than our Poisson nuclei.

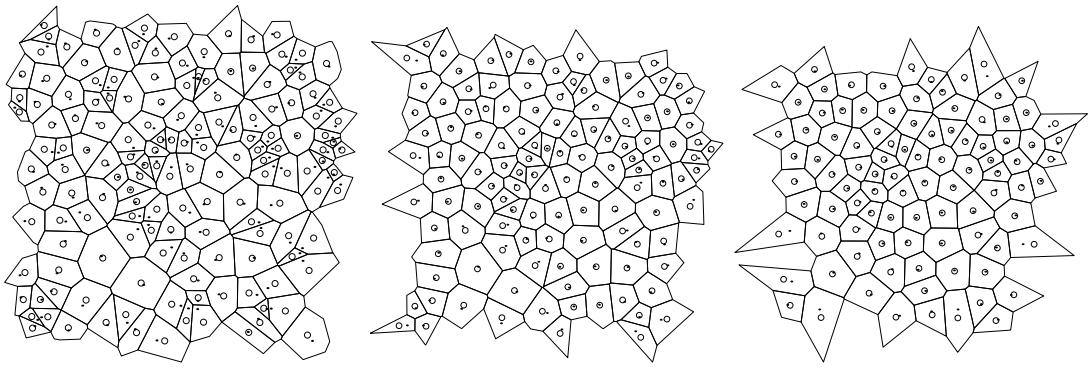
Figure 6.1 Centroid of the Voronoi tessellation.

Because the centre of gravity is a property which has more to do with the individual cells than with their neighbours, there may be real situations in nature where the c.g. of individual cells in a network has a role to play. For example, it is usually assumed in a mathematical Voronoi tessellation that all the nuclei stay fixed and do not move. But in a real situation this would imply that there

are some nuclei which lie very close to the walls. As no such nuclei would be stable, they more likely move away, and in that case the most reasonable prediction is that they move towards the centre of gravity of the cell.

Because they are stochastic in nature, computational VT could in theory take up any shape however awkward. Jafferali (1995), for instance, imposes constraints on the algorithm which creates VT's, namely in the course of the generation no new nuclei can assume a position too close to those of the existing ones. Thus there is an excluded volume within the network which develops throughout the generation process, and the Poisson point process does not over the whole space. Again, this excluded volume is relative to the neighbours of the cells rather than to themselves.

I suggest that an evolution centred around the centroids of a network can replace such constrint regarding the minimum allowable distance between two nuclei. And that this would be a more natural for a network to do because the nature of the procedure which makes it look inwards to itself, as opposed to being totally governed by its environment. An attractive feature of the movement of the nuclei towards c.g.'s is that the cells in effect takes into account not only themselves but also their surrounding, because, if nothing else, it was the latter who shaped their appearance in the first place. Therefore the nuclei would start to grow from a purely random position according to a Poisson point process. Then it would move towards the centroid of the cell, and as it does so the definition of its boundaries is also changed. The centroid is a stable position, as can be seen in Figure 6.2. Similar to the processes of covering, $\mathrm{C}^{\cdot}(\mathcal{V})$ introduced in § 6.3 on page ℵ, and dual Voronoi, $\mathcal{V}^n(\cdot)$ in § 6.3 on page 197, this centroid process can be recursive, that is $\mathcal{G}^n(\cdot)$. However, unlike the other two mentioned, this process very quickly becomes stable; Figure 6.2 (b) is much different from (a), whereas it is very similar to (c). The difference is in the size distribution as well as in the location of nuclei. The nonuniformity in the shape of a half-circle which can be seen in Figure 6.2 (a) is propagated to both Figure's 6.2 (b) and (c), but it has become much fainter than in the original.



(a) (b) (c)

Figure 6.2 *Nuclei evolution towards a centroid; (a) a Voronoi tesssellation based on the Poisson point process, $\mathcal{V}$, (b) the VT around its c.g.'s, that is $\mathcal{G}(\mathcal{V})$, and (c) the same process applied again to give the second order $\mathcal{G}^2(\mathcal{V})$.*

The program in § A.20 finds the centroid of a Voronoi network and $\mathcal{G}^n(\cdot)$.

There are three main steps in finding the centroid of a polygon. For each triangle of the triangulation of the polygon, first find the coordinates of its centroid, then its area. And then the centroid of the polygon is the weight average of the centroids of all the triangles. The first step involves finding the equation of two medians, which can be done by finding the mid point of an edge and then the slope to it from the opposite vertex. The second step can be carried out using the Heron's formula which finds the area by means of edge lengths.

Without loss of generality, let the polygon has n vertices at (a_i, b_i), where i increases from 1 to n in the clockwise direction. Assuming that all triangles which triangulate our polygon share one vertex at (a_1, b_1), so that they all are $\triangle_{ijk}$, where $i = 1$, $k = j + 1$ and the three vertices are in the clockwise direction (a_i, b_i), (a_j, b_j) and (a_k, b_k). There are three medians in each triangle, corresponding to the three vertices, but we only need two in order to solve for the centroid. Any two of them will give the same result, but in order to fix the algorithm let us choose those medians going out from the vertices i and j. Then, since there is no loop in the procedure, we can save the space by describing it here in linear steps as $c_i \leftarrow (a_j + a_k)/2$, $d_i \leftarrow (b_j + b_k)/2$, $m_i \leftarrow (d_i - b_i)/(c_i - a_i)$, and the first median has the equation $(y_1 - b_i)/(x_1 - a_i) = m_i$. Likewise the second equation can be obtained as $(y_1 - b_j)/(x_1 - a_j) = m_j$. Here the subscript of the centroid coordinates tells us in which

triangle it belongs. As the result of the above, we obtain for the k^{th} triangle $x_k = (a_i + a_j + a_k)/3$ and $y_k = (b_i + b_j + b_k)/3$.

The second step is to find the area. For this, we first find the edge lengths $s_{k1} \leftarrow (a_{ij}^2 + b_{ij}^2)^{1/2}$, $s_{k2} \leftarrow (a_{jk}^2 + b_{jk}^2)^{1/2}$ and $s_{k3} \leftarrow (a_{ki}^2 + b_{ki}^2)^{1/2}$, then the half-perimeter $s_k \leftarrow (s_{k1} + s_{k2} + s_{k3})/2$, and then the area $A_k \leftarrow (s_k(s_k - s_{k1})(s_k - s_{k2})(s_k - s_{k3}))^{1/2}$.

The third step solves for the polygonal centroid (x, y) as $x \leftarrow (\sum_k x_k A_k)/\sum_k A_k$ and $y \leftarrow (\sum_k y_k A_k)/\sum_k A_k$. Notice that the first step is parallel to the second- but not the third one. Moreover, finding c's and d's are parallel, as well as finding the various s_{ki}'s. So we could vectorise these using parallel processing. But on Matlab I think all the matrix operations are already vectorised, so we only need to put all the components to be run in parallel in a single matrix and then do all the operations at once as a single operation on the matrix, provided that this is possible.

The general algebraic solution to the above algorithmic procedure is obtainable by solving all the equations involved. This gives for any polygon the following theorem.

Theorem 6.1.

$$x = \left[\sum_{\kappa}(a_i + a_j + a_k)(a_i b_{jk} + a_j b_{ki} + a_k b_{ij})\right] \Big/ \left[3\sum_{\kappa}(a_i b_{jk} + a_j b_{ki} + a_k b_{ij})\right] \qquad (14)_{\text{vi}}$$

and

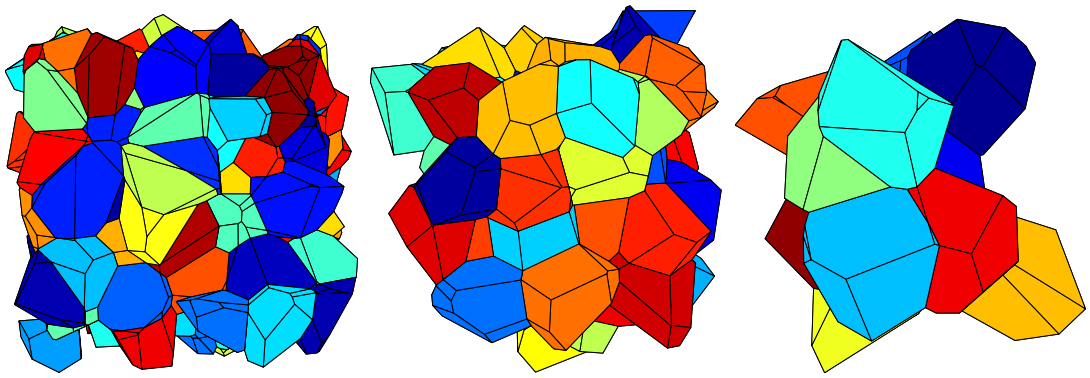
$$y = \left[\sum_{\kappa}(b_i + b_j + b_k)(a_i b_{jk} + a_j b_{ki} + a_k b_{ij})\right] \Big/ \left[3\sum_{\kappa}(a_i b_{jk} + a_j b_{ki} + a_k b_{ij})\right]. \qquad (15)_{\text{vi}}$$

Proof.: *Assuming that we accept the centroid of a triangle to be at the mean coordinates among its vertices, and the centroid of massive point bodies is their average coordinates weighted by their mass, then when $k = 1$ it is obvious by looking at the Equation's 6.14 and 6.15 that this theorem is true. Let us suppose that both these equations are valid for j triangles. We can present both of them in the form n/d where, necessarily, $n = \sum_\kappa x_i A_i$ and $d = \sum_\kappa A_i$. If we now add another triangle to this cluster, the new triangle will also be represented as a massive point and it will add the terms xA to the numerator and A to the denominator, making $n/d = (n + xA)/(d + A) = (\sum_{(J+1)} x_i A_i)/\sum_{(j+1)} A_i$. Now, x for a triangle $\triangle_{ijk}$ is $(a_i + a_j + a_k)/3$, and A calculated from Heron's formula above is $(a_i b_{jk} + a_j b_{ki} + a_k b_{ij})^{1/2}/2$, which yields Equation 6.14 and 6.15 for $\kappa + 1$. This proves both equations, and thus Theorem 6.1, by induction.* □

These general formulae, Equation's 6.14 and 6.15, cover the triangle itself, when we consider the latter as a polygon, and the solution when $k = 1$ in this case reduces to (x, y) being simply the average coordinates of the three vertices, which is the same as that we have earlier mentioned.

Earlier we have seen the evolution of the nuclei towards centroids in two dimensions. Let us look at the same thing in three dimensions. For this purpose, the program of § A.19, which is used in § 6.5 and explained by Algorithm 6.18 in page 202, is still inadequate, as its list of faces contains considerable amount of duplicates. This would have increased the resources required to do further work here. Therefore the program has been adjusted to that in § A.21 for the described in the following.

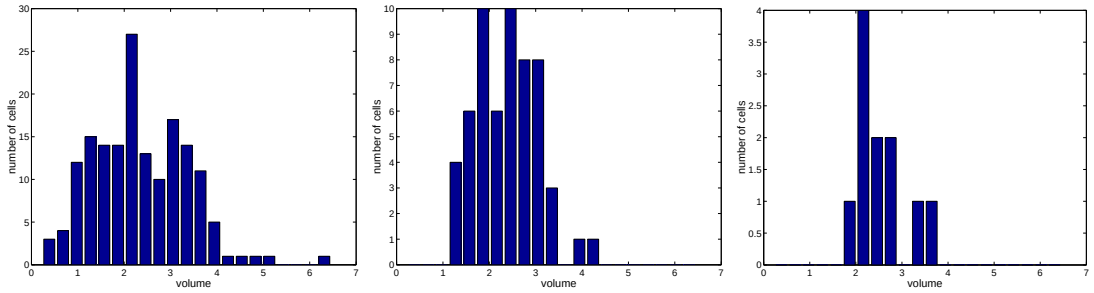
Similar to the idea shown in Figure 6.2, we draw in Figure 6.3 the VT's in three dimensions whose nuclei evolve towards the respective c.g.'s.



(a) (b) (c)

Figure 6.3 *Evolution of nuclei towards centroids in three dimensions, namely* (a) $\mathcal{V}^3$, (b) $\mathcal{G}(\mathcal{V}^3)$ *and* (c) $\mathcal{G}^2(\mathcal{V}^3)$.

This shifting in the nuclei positions preserves the mean of the size while cutting down its variance. The pictures in Figure 6.3 show how a 3-d VT transforms into its first- and then second order adjustment. If we plot the cell size distribution, we will come up with a picture similar to Figure 6.4. The appearance of the distribution where sizes are discretised into bins like this depends on the number of bins chosen.



(a) (b) (c)

Figure 6.4 *Change in size distribution as a result of a nuclei adjustment process,* (a) *the size distribution of a* $\mathcal{V}^3$, (b) $\mathcal{G}(\mathcal{V}^3)$ *and* (c) $\mathcal{G}^2(\mathcal{V}^3)$. *From a view point at 96° azimuth and 0° elevation.*

Figure 6.4 serves as a graphical presentation, but the statistics are the following. For our $\mathcal{V}^3$, $\mathcal{G}(\mathcal{V}^3)$ and $\mathcal{G}^2(\mathcal{V}^3)$ respectively the cell sizes are 0.9345 ± 0.4025, 0.9495 ± 0.2652 and 1.0315 ± 0.2086; the third central moment 0.0332, 0.0059 and 0.0061; the fourth central moment 0.0936, 0.0137 and 0.0036. These means are only meaningful when considered relatively with each other, since it depends on how one chooses the normalising basis. Here the basis chosen is the volume for the original Poisson point process, which is 1, divided by the original number of generators, which is 400. When we built this $\mathcal{V}^3$, 236 cells had been disposed of as they were thought to be those along the boundaries; when $\mathcal{G}(\mathcal{V}^3)$ a further 107 and finally when we built the $\mathcal{G}^2(\mathcal{V}^3)$ 46 more. This leaves us with 164, 57 and 11 cells remaining respectively in our $\mathcal{V}^3$, $\mathcal{G}(\mathcal{V}^3)$ and $\mathcal{G}^2(\mathcal{V}^3)$, all of which are shown in that order as Figure 6.4 (a), (b) and (c). Of course a great part of these are hidden behind others, so you can not possibly see them all there.

The sizes in Figure 6.4 are based on a volume magnification factor of 1,000, which is corresponds to the increase in size by a factor of three, but those in Figure 6.4 are based on a magnification 400, which corresponds to a factor of $(400)^{1/3}$ increase in the size.

As the new program has considerably changed from the old one, I list it again in § A.21 which has evolved from that which is listed in § ℵ. Unless I am mistaken, wherever the sign © appears in this monograph it means *copyleft* not copyright.

§ 6.4 Molecular dynamics

The Lennard-Jones function can be written $u(r) = -A/r^6 + B/r^{12}$, where A and B determine respectively the attractive and repulsive parts. Let the range parameter $\sigma = (B/A)^{1/6}$ and the energy parameter $\varepsilon = A^2/4B$, and the equation becomes $u(r) = 4\varepsilon\left[(\sigma/r)^{12} - (\sigma/r)^6\right]$. In other words, σ is the diameter of one of the atoms and ε is the well depth, *i.e.* energy constant. The force here is not $F = -\mathrm{d}u/\mathrm{d}t = (24\varepsilon/r^2)\left[2(\sigma/r)^{12} - (\sigma/r)^6\right]$, but $F = -\mathrm{d}u/\mathrm{d}r$.

Another approach in molecular dynamics simulation is to use numerical methods on the Newton's equation of motion. The most widely used is perhaps the Verlet algorithm, shown here as Algorithm 6.16. The initial values are r_0 and r_1.

Algorithm 6.16 *Verlet algorithm, L. Verlet (1967)*

for $i = 1$ to n **do**
  find f_i**;**
  $r_{i+1} \leftarrow 2r_i - r_{i-1} + f_i\Delta t^2/m + \mathrm{O}(\Delta t^4)$ **and**
  $v_i \leftarrow (r_{i+1} - r_{i-1})/(2\Delta t) + \mathrm{O}(\Delta t^2)$**.**
endfor □

There are variants and modifications of Algorithm 6.16, for example the Leapfrog Verlet algorithm which has three steps instead of two, that is $v_{n+1/2} = v_{n-1/2} + f_n/m\Delta t + O(\Delta t^3)$, $r_{n+1} = r_n + v_{n+1/2}\Delta t + O(\Delta t^4)$ and $v_n = (v_{n+1/2} + v_{n-1/2})/2 + O(\Delta t^2)$; or the velocity Verlet algorithm where $r_{n+1} = r_n + v_n\Delta t + f_n\Delta t^2/(2m) + O(\Delta t^3)$ and $v_{n+1} = v_n + \Delta t(f_{n+1} + f_n)/(2m) + O(\Delta t^3)$, or in the form normally used in practice in which $v_{n+1/2} = v_n + f_n\Delta t/(2m)$, $r_{n+1} = r_n + v_{n+1/2}\Delta t$ and $v_{n+1} = v_n + f_{n+1}\Delta t/(2m)$.

The van der Waals force is an attractive force acting between molecules. In the case of gases, each molecule consumes some space, so the dynamic volume is less than overall volume by an amount bn when b is a constant and n the number of molecules. Here b is $N_A v$, where v is the volume slightly larger than the volume of each molecule of gas and N_A the Avogadro number, $N_A = 6.022 \times 10^{-23}$ mol^{-1}. The pressure of gas we see is the pressure of gas detected, which is less than the real pressure by F/A, where $F = \sum F_i$ and $F_i = \sum f_j$. Here i runs from 1 to n and j from 1 to $(n-1)$ in the same set of molecules. The reduction of the total force comes from all molecules, but this reduction from each molecule is in turn affected by those molecules around it. Therefore $i \neq j$ and j runs from 1 to $(n-1)$ as mentioned above. Suppose that each molecule attracts another molecule by a force a. Then, in a given volume V, $F = \sum F_i = \sum_i \sum_j f_{ij} = \sum_i (n-1)a/V = n(n-1)a^2/V^2$. For very large n we can say that $(n-1) \approx n$, and therefore $(P + an^2/V^2)(V - bn) = nRT$.

The nature of this mutual attraction between molecules, which reduces the pressure in gas, comes to light in the case of solids. The distribution of charges around a neutral atom in solid fluctuates in the time scale $\tau < 10^{-16}$ s. This charge imbalance makes each atom behave as an electric dipole. This electric dipole has an electric field around it, which affects other atoms nearby and binds the two together.

The electric dipole moment is $p = qa$, where a is a vector from the negative to the positive charge. The electric field around a charge q_1 has a magnitude $E = q/(4\pi\varepsilon_0 r^2)$. It acts on a nearby charge q_2 with a force of magnitude $F = u = q_2 E = q_1 q_2/(4\pi\varepsilon_0 r^2)$, where u is the potential energy of the two charges. This force itself is the Coulomb force, and the electric potential around q_1 is u/q_2, that is $V = q_1/(4\pi\varepsilon_0 r)$.

For an electric dipole, $V = (1/(4\pi\varepsilon_0))(q/r_1 - q/r_2) = q(r_2 - r_1)/(4\pi\varepsilon_0 r_1 r_2)$. Since $a \ll r$, a being of the order of the atomic size, $r_1 r_2 \approx r^2$. Let θ be the angle between a and r. Then, also since $a \ll r$, we have $r_2 - r_1 \approx a\cos\theta$. Therefore we now have $V \approx qa\cos\theta/(4\pi\varepsilon_0 r^2) = p\cos\theta/4\pi\varepsilon_0 r^2$ and consequently $E_r = -\partial V/\partial r = 2p\cos\theta/(4\pi\varepsilon_0 r^3)$ and $E_\theta = -\partial V/(r\partial\theta) = p\sin\theta/(4\pi\varepsilon_0 r^3)$.

We neglect E_θ for long-range interaction, and simplify E_r to $E_r = a/r^3$. We shall hereafter use the terms *dipole* and *atom* interchangeably. If this electric field is produced by atom p_1, it could induce in another atom $p_2 = \alpha E_r$, where α is the molecular polarisability of the second atom. This second atom will have an energy of interaction with the first one, $u = -p_2 E_r = -\alpha E_r^2 = -\alpha a^2/r^6$. This force is attractive.

The repulsive force is more complicated, and is generally thought to arise from the Pauli exclusion principle and the coulombic repulsion of the electrons in the outer orbit. It is generally assumed to be either $u \approx A/r^{12}$ or $u \approx a\exp(-r/\rho)$, where ρ is a range parameter (*cf* de Podesta, 1996).

The Lennard-Jones potential is often written as $u = -4\varepsilon\left[(\sigma/r)^6 - (\sigma/r)^{12}\right]$, where σ is a range parameter that indicates the approximate size of an atom and ε an energy parameter that indicates the strength of the interaction between atoms. In other words, it is the minimum value of u.

The Lennard-Jones potential between two atoms has the minimum value $u = \varepsilon$ at $r = 1.1225\sigma$. Figure 6.5 is a plot $y = -4(1/x^6 - 1/x^{12})$, where $y = u/\varepsilon$ and $x = r/\sigma$. The force which the potential curve of Figure 6.5 acts on another particle is shown in Figure 6.6. The force produced by a dipole is calculated from $F = -\mathrm{d}u/\mathrm{d}r = 24(\varepsilon/\sigma)\left[(\sigma/r)^7 - 2(\sigma/r)^{13}\right]$. But in our units $y = u/\varepsilon$ and $x = r/\sigma$ above the equation becomes $y = 24(1/x^7 - 2/x^{13})$, which is plotted in Figure 6.6.

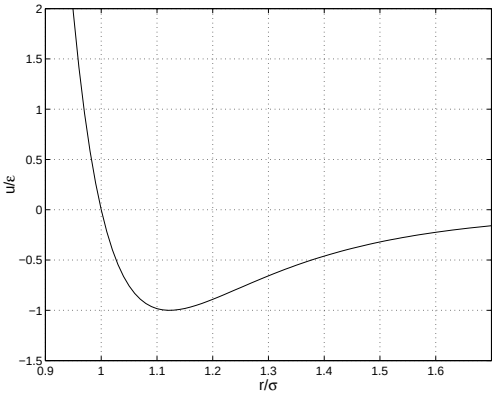


Figure 6.5 Lennard-Jones potential.

Figure 6.6 shows that the force is zero at the distance $x = 1.1225\sigma$ away from the particle. Closer than this point the repulsive force increases rapidly towards infinity. Further away from this point, however, the attractive force increases to a maximum and then gradually dies down. This maximum force occurs when $\mathrm{d}x/\mathrm{d}y = 0$, that is at $x = 1.2445\sigma$.

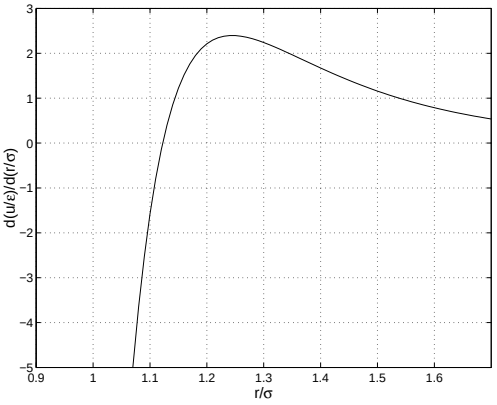


Figure 6.6 Force corresponding to the Lennard-Jones potential.

In gases, the molecules travel past one another so fast that they hardly notice the wells of negative energy surrounding other molecules, let alone stop and rest there. But even here the attractive force produced by these wells is probably what give rise to the term an^2/V^2 in the van der Waals equation. Solid molecules, in contrast, hardly have kinetic energy and therefore prefer to sit in such wells of their neighbours, which is the reason why solids hold and neither flow as liquid does nor disperse like gas.

The most cohesive structures of solids are crystals, where the Lennard-Jones potential culminates in a minimum cohesive energy of the lattice. This cohesive energy is $U = N_A u_i/2$. Here u_i is the summation over all pair potential energies of Lennard-Jones type, $u_i = \sum_{i\neq j}^{N_A} u(r_{ij})$. Let $r_{ij} = \alpha_{ij} r_0$, where r_0 is the nearest neighbour distance. Then $U = -2\varepsilon N_A\left[(\sigma/r_0)A_6 - (\sigma/r_0)A_{12}\right]$, with the lattice sums being $A_6 = \sum(1/\alpha_{ij}^6)$ and $A_{12} = \sum(1/\alpha_{ij}^{12})$. These lattice sums are calculated from the network structure in terms of infinite series that converge very quickly.

To find the optimum value of σ/r_0 we shall let $b = \sigma/r_0$ and write differentiate $\mathrm{d}U/\mathrm{d}b = -2\varepsilon N_A(6b^5 A_6 - 12b^{11}A_{12}) = 0$. As a result, $r_0 = \sigma(2A_{12}/A_6)^{1/6}$, the cohesive energy per mole $U = -(A_6^2/2A_{12})N_A\varepsilon$ and the cohesive energy per molecule $u = U/N_A = -(A_6^2/2A_{12})\varepsilon$.

§ 6.5 Problem definition and algorithms

Walls in a 3-d Voronoi tessellation are isomorphic to bonds that link between its cells. Therefore we can redefine the problem of a particle passing through the hole in a wall, into the cell chamber and out through a hole in another wall, and so on, as that of the passage of a mathematical particle through a tube that links between two cells, into a cell and out through one of the other tubes. In reality the solid parts which make up the partitions have thickness, therefore the holes through the walls will have a nonzero thickness and the volume of the cell chamber is smaller than that of the otherwise mathematical Voronoi cell.

Each vertex of a Voronoi polygon has three walls of the latter attached to it. I find the centre of gravity† of thess three faces and link them together. This truncates the coigns and produces for each polygon its dual self. The next step is to link together the midpoints of all those edges of the polygon that have a vertex in common. The edges of this last polygon bound and define the void volume of the cell chamber.

Next, the cross section of the hole within each wall is taken to be the area on the original wall which is bound on all sides by the second order covering lattice of that wall.

The filtering membrane in this case is assumed to be isomorphic and homogeneous. The size of the particles is taken to be reasonably smaller than the void in general, so that the attraction between particles and the adsorption to the wall play a more prominent role than the physical blockages by individual particles. The particles are all assumed to be of the same size.

Algorithm 6.17 is an algorithm to do filtration that is being developed. Here N contains the neighbourhood information

Algorithm 6.17 *Filtration in Voronoi tessellation*

$(v_a, c_a) \leftarrow$ **find** Voronoi tessellation;
$d_l \leftarrow$ **find** Delaunay triangulation;
$N_c \leftarrow$ **find** cell neighbours;
$c \leftarrow c_a$ which lie completely within $[0,1]^3$;
$v \leftarrow v_a$ which belong to some c;

□

My work on filtering membranes results in another Voronoi algorithm and program which is different from the programs listed in §'s A.3 and A.4. Because it is also written anew from scratch, this new algorithm has nothing to do with the previous two as regarding the data structure and the logic of its method. It is given in § A.19.

The current convention for variables is this. A single alphabet or entity means a list, for example `c` is the cell list and `va` is the list of all vertices original created. Another example is b, a structure for bonds which contains the list of the cells of bonds, the number of vertices of the face represented by each bond, the list and the map of these vertices, the ordered list of the vertices of each face and another list similar to this but cyclic, and the number of cells connected to each bond. Similar to the structure of `b` is that of *bdr*, or b_d in Algorithm 6.18, which is a list of the bonds along the border, each of which is connected to only one cell.

Two entities xy makes x of y, for instance `vc` is vertices of cells, which is a structure that contains the number of vertices of each cell and the list and the map of all these vertices. I try to preserve the space by creating a short but easy to understand naming convention. Also, the lines are put together, which means that the number of lines of the codes is more than what appears here, some of the lines having six logical lines or more to them. But the structure of the program still remains intact, therefore it should be possible without much difficulty to compare the program in § A.19 with Algorithm 6.18 which explains it.

Three entities xy^2 or y_x^2 are $y \times y$ matrix of x, where the latter relates two entities of the former, for example `bcc` is the bonds mapped on to a cell matrix. In the algorithm, v_n and c_n are respectively the `vin` and `cin` in the program, the `vn` and `cn` there being verbosely described in the algorithm as the *number of* v and c.

Algorithm 6.18 *Voronoi data structure for the study of membrane filters.*

$(v_a, v_c^a) \leftarrow$ **find** a 3-d Voronoi tessellation;
$v_n \leftarrow$ index of $0 < v_a < 1$;
$c_n \leftarrow$ index of c all the v's of which are in v_n;

† aka centre of mass, centroid.

$v \leftarrow v_a \in v_n$;
$v_c \leftarrow v_c^a \in c_n$;
$t_a \leftarrow$ **find** a 3-d Delaunay tessellation;
for all $t_i \in t_a$ which are connected to two cells **do**
  $t \leftarrow t_i$;
else
  $b_d \leftarrow t_i$;
end
$b \leftarrow c_b^2 \leftarrow t$;
$b \leftarrow b_d$;
find Delaunay triangulation in $(3-1)$ dimensions for all faces;
order the vertices of each face; □

§ 6.6 Simplified algorithm for filtration

The first algorithm we shall now develop is a reasonably simplified one. But it turns out to my surprise that this simple algorithm links us to the continuum percolation of hard spheres or even of particles with irregular shapes. This is because of the assumptions that we shall make.

One such assumption is that the variance of the filter cell size distribution is small. For this, Jafferali (1995) would probably have applied his favourite constraint code on a Voronoi tessellation. But I have my own approach which I think is more logical, namely by using the operator $\mathcal{G}(\cdot)$ on a $\mathcal{V}$ (*cf* § ℵ), which we shall use for now.

The second assumption is that the particle size is reasonably smaller than the size of the smallest void of the original system. This is the very assumption which I think seems to link us somehow to the percolation of spheres in continuum, the space considered being that of each void.

Also, we assume that the particles are attracted towards one another and towards the walls by various interactions which may include the van der Waals force and the force from electrostatic interactions.

Let θ be the angle that the line from the c.g. of a each cell to the mid point of a face makes with the horizontal plane. Then another possible assumption is that all particles prefer a trajectory which goes through a face which has the maximum θ.

The third assumption is that the volume of each membrane pore is 80 per cent the volume of the corresponding Voronoi pore. The fourth assumption, blockages due to the clustering caused by the attrition forces between particles can occur within pores. This excludes cake formation and blockage at entrances to, or exits from the pore.

The last assumption above is similar to assuming that clustering due to attrition can occur in stagnation regions. Since the flow of the suspension reaching the membrane is strong, *i.e.* the pressure high, there can be no clustering there. The flow from one pore into another is also faster than the flow within one. This is amount to assuming that the effective diameter of the hole within each face of a pore is considerably smaller than its diameter.

Upon reaching the membrane, each particle in the troop seeks out the bond closest to it, and begins its journey through the membrane. Within the membrane, it always follow the bond which has the greatest gradient available. Therefore the top bonds should map to the bottom ones one to one. But because the possible blockages during the course of operation, this may not be so and the mapping is instead one to many.

First we consider the case in two dimension, so volumes becomes areas and the volumetric flow rate the distance travelled. In our discrete time, if v is the flow velocity, n the number of particles, ρ_v the total volume ratio of the particles and r their radius, then we have $n\pi r^2 = v\Delta t$. Assuming that $v = 1$ ms^{-1} and $r = 100$ μm, then $n = 10^3\rho_v/\pi$ or approximately $318\rho_v$ particles for each time step.

Next we will study the suspension in a square box in order to find out ρ_v. Because we shall assume that the particles flowing through the pores percolate as though there is no flow, we will consider here the suspension that is simply contained within our box without moving about. I feel that the assumption that particles could percolate that way in pores is justified since the flow is in steady state, protected from the turbulence outside by all the solid structures making up the walls of the membrane.

The percolation program first generates random positions of the particles, then moves apart those which are too close together so that in the end they only touch each other. Particles which are

separated by a distance less than $1.4r$ move towards each other until they touch. Lastly, touching particles never separate.

We know that the total volume ratio of the particles has the upper limit of 0.9069, because that is the density of the closest packing of circles on a plane.

Packing circles on the plane becomes densest when the circles are arranged as a hexagonal lattice with the packing density of $\pi/2\sqrt{3}$. Packing of spheres is similarly at its highest density if the arrangement is that of the face-centred cubic lattice with the packing density $\pi/3\sqrt{2}$.

§ 6.7 Filtering problem when physical blockage is prominent

Assuming particles to be spherical, the size of particles to be constant, and that this size is compatible with the size of the holes in the walls and the voids so that physical blocking is responsible for most of the blockages in the membrane. As in § 6.7, the hole in each wall is taken to be the polygon which results from recursively finding a covering polygon for the face twice.

Redefine the problem of particles' passage through walls and voids as that of particles travelling along edges of the dual lattice of the Voronoi structure, *i.e.* the Delaunay triangulation. Each of these edges corresponds to a face in the original physical lattice. To each edge is thus ascribed the details of the cross section of the hole through that face, namely the shape and the dimension, as well as the gradient it makes with the horizontal plane. Upon reaching a vertex, the particle ball will choose the next path, *i.e.* bond in the dual lattice, which has the maximum gradient to pass through. However, if this bond is too small or if it is blocked, the particle with choose from among the remaining paths the one which has the maximum gradient, and so forth. If the size of the particle is approximately that of the hole it tries to pass through, within two per cent of the latter, say, then the particle will blind the passage at that point. The difference between blinding and blocking is in the degree of tightness that the particle sits in the hole, which reflects in the degree of difficulty to remove it by backflushing. This degree is not constant but a function of the relative size between the two parties involved, *i.e.* the particle and the hole.

§ 6.8 Percolative filtering with very small particles

At this point a new filtering algorithm is considered and investigated. We introduce an assumption that the particles are all of the same size which is very small compared with the size of the pores. So there is neither cake formation nor blinding by a single particle. Assume that the solid particles suspended in a fluid medium, being dragged downwards under their own weight.

Assume that the van der Waals force plays a signigicant part and, similarly to § ℵ, that these interactions between particles are governed by the Lennard-Jones equation. Assuming that each particle is spherical and acts as a dipole with the electric dipole $p = 10e$, where e is the electric charge on a proton, $e = 1.602 \times 10^{-19}$ C. Then (*cf* de Podesta, 1996)

$$E_r = \frac{p}{4\pi\varepsilon_0 r^3}, \qquad (16)_{\text{vi}}$$

and the energy of interaction between two particles becomes

$$u = -\alpha E_r^2 = \frac{\alpha p^2}{(4\pi)^2\varepsilon_0^2 r^6}, \qquad (17)_{\text{vi}}$$

where α is the molecular polarisability of one particle under the influence of the other. If we assume that the repulsive force acts in such a way that the repulsive energy is $u = c/r^{12}$, where c is a constant, the Lennard-Jones potential becomes Equation 6.18.

$$u = -\frac{\alpha p^2}{(4\pi)^2\varepsilon_0^2 r^6} + \frac{c}{r^{12}}. \qquad (18)_{\text{vi}}$$

Comparing Equation 6.18 to the form

$$u_r = -\frac{A}{r^6} + \frac{B}{r^{12}}, \qquad (19)_{\text{vi}}$$

we have $A = \alpha p^2/((4\pi)^2\varepsilon_0^2)$ and $B = c$. Or if we compare it to the form

$$u_r = -4\varepsilon\left[(\frac{\sigma}{r})^6 - (\frac{\sigma}{r})^{12}\right], \qquad (20)_{\text{vi}}$$

then we have $\sigma = (B/A)^{1/6} = (16c\pi^2\varepsilon_0^2/(\alpha p^2))^{1/6}$ and $\varepsilon = (A^2/4B) = \alpha^2 p^4/(4c(4\pi)^4\varepsilon_0^4)$. Notice that here σ and ε are simply parameters, not the Stephan-Boltmann constant and the dielectric

constant, σ is a range parameter and approximates the size of an atom while ε is an energy parameter which shows the strength of the interaction between particles. At $r = \sigma$ the value of u_r is zero, whereas u_r has the minimum value of $-\varepsilon$.

The force between two particles is Equation 6.21.

$$F = -\frac{\mathrm{d}u}{\mathrm{d}r} = -\frac{6\alpha p^2}{(4\pi)^2\varepsilon_0^2 r^7} + \frac{12c}{r^{13}}. \qquad (\mathbf{21})_{\mathrm{vi}}$$

At the equilibrium separation, r_0, $F = 0$ and therefore Equation 6.21 yields the minimum distance in Equation 6.22,

$$r_0 = \left(\frac{32c\pi^2\varepsilon_0^2}{\alpha p^2}\right)^{\frac{1}{6}}. \qquad (\mathbf{22})_{\mathrm{vi}}$$

The equation for the minimum energy is obtained by substituting r_0 from Equation 6.22 into Equation 6.18, which gives

$$u = -\frac{\alpha p^2}{(4\pi)^2\varepsilon_0^2 r^6} + \frac{c}{r^{12}} \qquad (\mathbf{23})_{\mathrm{vi}}$$

$$= \frac{1}{r_0^6}\left(-\frac{\alpha p^2}{(4\pi)^2\varepsilon_0^2} + \frac{c}{r_0^6}\right) \qquad (\mathbf{24})_{\mathrm{vi}}$$

$$= \frac{1}{r_0^6}\left(-\frac{\alpha p^2}{(4\pi)^2\varepsilon_0^2} + \frac{c\alpha p^2}{32c\pi^2\varepsilon_0^2}\right) \qquad (\mathbf{25})_{\mathrm{vi}}$$

$$= -\frac{\alpha p^2}{32\pi^2\varepsilon_0^2 r_0^6}. \qquad (\mathbf{26})_{\mathrm{vi}}$$

Returning to Equation 6.21, the attractive force has the maximum value at the point where $\mathrm{d}F/\mathrm{d}r = 0$. Differentiating F in Equation 6.21 with respect to r, we arrive at Equation 6.27.

$$\frac{\mathrm{d}F}{\mathrm{d}r} = \frac{42\alpha p^2}{(4\pi)^2\varepsilon_0^2 r^8} - \frac{156c}{r^{14}} \qquad (\mathbf{27})_{\mathrm{vi}}$$

From this the distance where this maximum force occurs is given by Equation 6.28,

$$r_m^6 = \frac{156c(4\pi)^2\varepsilon_0^2}{42\alpha p^2} = \frac{416c\pi^2\varepsilon_0^2}{7\alpha p^2}. \qquad (\mathbf{28})_{\mathrm{vi}}$$

Then by putting Equation 6.28 into Equation 6.21 we have this maximum force in Equation 6.32.

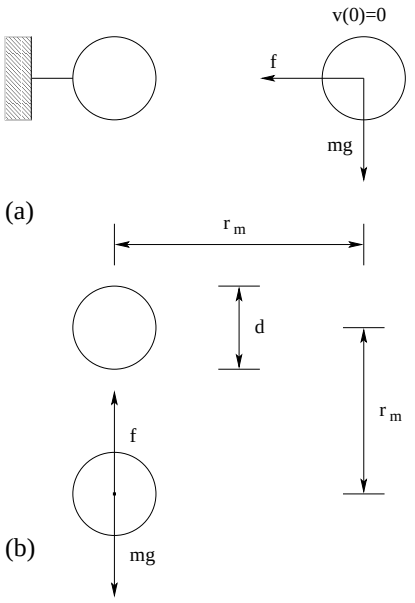
$$F_m = -\frac{6\alpha p^2}{(4\pi)^2\varepsilon_0^2 r^7} + \frac{12c}{r^{13}} (\mathbf{29})_{\mathrm{vi}}$$

$$= \frac{1}{r^7}\left(\frac{-6\alpha p^2}{(4\pi)^2\varepsilon_0^2} + \frac{12c(42)\alpha p^2}{156c(4\pi)^2\varepsilon_0^2}\right) (\mathbf{30})_{\mathrm{vi}}$$

$$= -\frac{9\alpha p^2}{52r^7\pi^2\varepsilon_0^2}.(\mathbf{31})_{\mathrm{vi}}$$

$$(\mathbf{32})_{\mathrm{vi}}$$

Let each of the spherical particles has a radius $r = 1$ μm and the density $\rho = 3{,}000$ kg $\cdot$ m^{-3} (compare the density of silicon, which is 2,329 kg $\cdot$ m^{-3}). Then the mass of the particle is $m = \rho V = 3000 \times (4/3)\pi \times (10^{-6})^3 = 1.26 \times 10^{-14}$ kg and the weight is $mg = 1.26 \times 10^{-14} \times 9.8 = 1.23 \times 10^{-13}$ N.



We may now find the attractive distance, the maximum distance whereby two particles will come together under the van der Waals force. Figure 6.7 shows the capturing of one particle by another when one particle is fixed in space and another particle has zero velocity but is free to move. Recall that the capture occurs when the accelaration due to the weight of the particle g equals the acceleration a due to f.

Figure 6.7 Force balance with one particle fixed.

If we suppose that our particle have the same polarisability as that of a benzene in its gaseous state, C_6H_6, *i.e.* $\alpha = 11.61 \times 10^{-40}$, then in this case $\alpha = 11.61 \times 10^{-40}$ $F^{-1}m^4$ (*cf* de Podesta, 1996) and Equation 6.32 gives us the separating distance between the two particles which gives the maximum attractive van der Waals force, which is 444 μm. Giving our particle other values of α, with the value of α for methanol gas CH_3OH, *i.e.* $\alpha = 3.860 \times 10^{-40}$ $F^{-1}m^4$, we have this distance $r = 379$ μm, and with $\alpha = 1.647 \times 10^{-40}$ $F^{-1}m^4$, that of water vapour, the distance becomes $r = 336$ μm.

These values are rather large compared with the radius of the particles, therefore I shall opt instead to a bigger size of particles, when $r = 5$ μm. With the radius of five microns, the particle has a mass of 1.571×10^{-12} kg and its weight becomes 1.64×10^{-11} N. Then if we adopt the value of α for water vapour, $\alpha = 1.647 \times 10^{-40}$ $F^{-1}m^4$, then from Equation 6.32 the distance where the force is maximum becomes $r = 167$ μm.

But this is not everything. So far we have only considered what two particles will do when one of them is fixed and the other one has no initial velocity. There are two other things that can happen, the particles may both be moving down beside each other under gravity and the path of the particle which is effected by their mutual attraction. Even when two particles do not come together, their path can be deviated by the van der Waals force. But here for simplicity we shall neglect this and assume that particles either come and stay together or they experience no mutual force whatever. Only if they come together will their final path and velocity be affected, and these depend on their combined momentum.

When particles move relative to each other, their captive velocity depends on their relative velocity. They tend to join each other more easily if their paths are along side each other and goes in the same direction. In the extreme case where both particles move in the same direction with zero relative velocity, they would come together across a vast gap in between indeed, had there been no frictional force due to viscosity that acts to drag them.

In the present study, the friction due to the fluid in the medium is neglected, together with the relative velocity between the particles, for the purpose of deciding whether particles will come together. It is expected that on average particles coming to within the capturing radius of each other will come together. This radius is larger than the radius of maximum force in Equation 6.32 above. In other words, we expect our particles to behave like solid spheres with a well defined boundary for their sphere of influence and a much simplified capturing mechanism.

We shall define this radius of captivity, r_c, to be one third that of the radius of maximum force r_m. From Figure 6.5 this is the point where $r_c = 1.5\sigma$, and because r_m is here 1.2445 we have $r_c/r_m \approx 1.2$. For Figure 6.5, this is the same as saying that $r_c/r_0 = 1.34$.

One additional point is that, in real situation when particles form a cluster their collective value of the molecular polarisability will change, and this value will not be the same for the cluster

as it is for the individual particles. But here we assume that they are the same, which means that the capturing radius of a cluster will be the same as that of the particles which form it.

It may worth mentioning here that the molecular polarisability is related to an optical property, *viz.* the refractive index n_r, by the relationship $\alpha = \varepsilon_0(n_r^2 - 1)/n$, where n is the number density of molecules. For gases $n = P/(k_B T)$, whereas for liquids $n = N_A \rho/m$, m being the mass.

Next we shall concern ourselves with clusters of particles thus formed. All clusters will be assumed to be a closest-conglomerate of particles which form them, with the densest packing density possible in three dimensions, that is $\pi/3\sqrt{2}$ of the face-centred cubic lattice. Numerically this is 0.7405.

Notice that the packing density in two dimensions can be higher than this. Packing circles on a plane is the densest of all with its packing density of $\pi/2\sqrt{3}$.

Shown in Figure 6.8 are the hexagonal lattice packing in two dimensions and the cubic close pack. In the cubic close packing spheres in every third layer lie vertically straight on top of one another. Each face of its cubic section looks like the second picture in Figure 6.8. Similar to the cubic close packing is the hexagonal close packing spheres in every alternate layer of which lie over one another. Both the cubic- and the hexagonal close packing have the same packing density which, in the case of the former, is calculated, from the second picture, as $\rho = \sum v_s / \sum v_c$, where v_s are the total volume of sphere segments in the unit cell which has the volume v_c. Here $v_c = (2\sqrt{2}r)^3$ and $\sum v_s = (8(1/8) + 6(1/2))4\pi r^3/3$. In two dimensions, the densest packing is calculated from the first picture. The density of the packing is $\sum a_c/a_r$, where a_c are the areas of the circles and a_r is the area of a rectangle. These two areas are namely $a_c = 2(\pi r^2)$ and $a_r = 2r(2\sqrt{3}r)$, which give the density $\rho = \pi/2\sqrt{3}$.

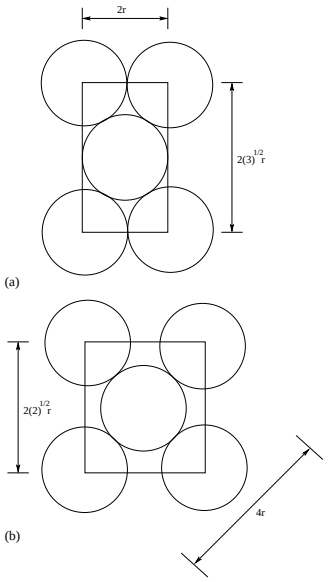


Figure 6.8 The packing density calculation of the closest-packed densities.

§ 6.9 Percolation within percolation

Particles suspended in a fluid generally have their diameter d_p much smaller than the pore diameter d_v of a filtering membrane. Yet these small particles can cause blockages of the membranes due to attrition among themselves. Since in this case $d_p \ll d_v$, blockages due to blocking or blinding of the individual particles (*cf* Jackson, 1994; Jafferali, 1995) become out of question.

Having investigated both the percolations of networks and continuum, I suggest that the blockage of these smaller particles in membranes is due to a double percolation phenomena, one the percolation of the suspension continuum, the other the percolation of the centroidal Voronoi network. As a reminder of a centroidal Voronoi network, it is a Voronoi tessellation on generator points which are the centroids of a Voronoi network which either is generated from Poisson point generators or is another centroidal Voronoi network.

Because percolation is a study of the behaviour of two phases, and because in general $p_c \neq 1/2$, there are no less than three regions of behaviours to consider in each percolational investigation instead of two (*cf* Tiyapan, 1997). When $p_c < 0.5$, these three regions are $p < p_c$, $p_c \leq p \leq (1 - p_c)$ and $p > (1 - p_c)$. When $p_c > 0.5$, they are $p < (1 - p_c)$, $(1 - p_c) \leq p \leq p_c$ and $p > p_c$. The case where $p_c = 0.5$ is assumed to be very rare in nature, and so can be neglected in the present study.

When a suspension becomes so high that the average interparticle distance has become such that the attrition due to van der Waals force is prominent, the suspension will solidify into a moisted bed of particles. According to the percolation theory, we may define the point where this spontaneous solidification occurs to be that point where there is a single cluster, under a mutual van der Waals force, which traverses the whole continuum in a certain well-conditioned direction, that is to say, a direction which may represent the diameter of the network. Furthermore, let us call a critical concentration ρ_c the minimum concentration at which this infinite cluster appears.

Then we have for our suspension a continuum percolation with three regions of behaviour similar to those we have found in the case of network percolation. Furthermore we map the space of ρ_c on to that of $0 \leq p_c \leq 1$, where $p_c = 0$ means there are no particles suspended in the fluid, in other word $\rho = 0$, and $p_c = 1$ is where the suspended particles form a bed in the closest packed structure, that is $\rho = \rho_{\max}$. We also assume, without the loss of generality, that $p_c > 0.5$. Then we have the following as the three regions, $p < (1 - p_c)$, $(1 - p_c) \leq p \leq p_c$ and $p > p_c$.

We are interested in neither the cases $p < (1 - p_c)$ nor $p > p_c$, since the former implies too dilute a concentration for the attrition due to the van der Waals force to cause an infinite cluster, while the latter means that the suspension is so concentrated that they solidify instantaneously, simultaneously in all pores.

When $(1 - p_c) \leq p \leq p_c$, all pores has an equal probability of being solidified, and so the per cent total solidified cells now depends on the topology of the network,in this case Voronoi, and the probability where the critical phase change occurs becomes the critical probability of the network.

In the case of dead-end filtration the flow is in one direction, therefore the critical flux reduction which is the result of percolation of the network occurs at the point where the cross section of the network, not the network itself, percolates. This is because such percolation in the cross section in the plane perpendicular to the flow direction will cause a bottle neck in the flow and therefore determines the flux. This phenomenon is summarised in Figure 6.9.

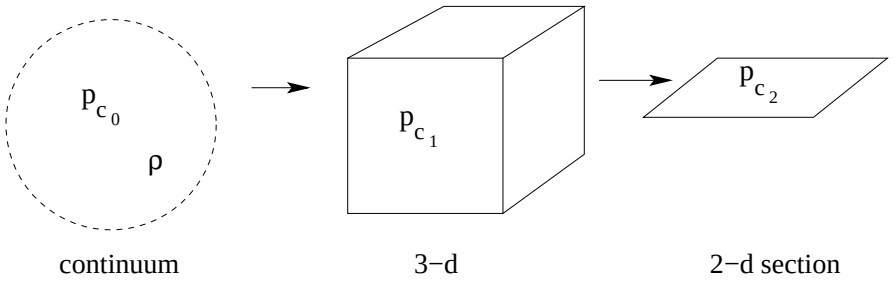


Figure 6.9 *Percolation within percolation;* $0 \leq \rho_0 \leq \rho_{\max}$, $0 \leq p_0 \leq 1$, $p_c = p_{c_2} \cdot p_0$.

Next we must define the critical probability of the overall system. Since the system involves two kinds of probability, that is continuum and network, and assuming p_{c_0} and p_{c_2} are two independent probabilities, then we have the overall probability is

$$p_c = p_0 p_{c_2}, \qquad (\mathbf{33})_{\text{vi}}$$

where $(1 - p_{c_0}) \leq p_0 \leq p_{c_0}$.

If our membrane is homogeneous, the reduction in the area perpendicular to the flow becomes $\Delta A = A_v p_{c_2}$, where $V_v/V_t = A_v/A_t$, V_v and V_t are respectively the void volume and the total volume of the membrane, and similarly for the areas A_v and A_t.

§ 6.10 The first part, suspended particles

Because of the complex nature of the problem, there is no single algorithm but rather there is an algorithm for each job. The first task is to study the percolation of the spherical particles under the van der Waals force.

Continuing from the development in § 6.10, particles are spherical in shape with $r_p = 5$ μm and the capturing radius is $r_c = 167 \times 1.2 \approx 200$ μm. First we shall study particles within a cubic box of side length 2 mm. Particles start as a suspension with no obvious velocity. They stick together and to the walls.

When particles come together, they form a porous globule, having the densest packing density of the hexagonal or cubic close packing. When this happens, we discard the individual particles and consider instead the globular cluster which they formed. The cluster is porous, so its new radius is $r = (\pi/3\sqrt{2})(3\sum v_i/4\pi)^{1/3} = (\pi/3\sqrt{2})(3nv/4\pi)^{1/3} = \pi^{2/3}(nv)^{1/3}/(3^{2/3}2^{7/6})$.

But we do not know the rules by which these particles stick themselves together, whether they form a closest-packed globule or some other shapes. It is quite certain that whatever shape they are after, they may not retain it for long because there is a limited space within each pore that will put constraints on the way they grow. I shall call the growth of clusters into globules mentioned above by the name globular formation.

Other clustering mechanisms possibly include what I shall call the tetrahedra formation. By this I mean that each one of our spherical particles attaches itself to three other particles, forming a tetrahedron whose side lengths are two times their radius. The next free particle may fit into any of the available attachment sites, *i.e.* the free triangular faces of an existing cluster. We may suppose that it always choose the closest one among such sites if this is available; if not, then the next closest one and so on.

These are only two among all the possiblities, namely the globular and tetrahedra formations. There could well be others, as well as a mixture of them. On the other hand, each material of which the particles are made may decide the particular cluster shape it prefers. Detailed analyses in thermodynamics and quantum mechanics are needed if we were to understand this cluster formation in continua under spatial constraints. Neither of these is within the scope and time constraint of the present work, though both of them merit a detailed investigation which I plan to carry out in the future.

Another problem arises when we come to consider percolation of our membrane. We may, for instance, say that it percolates when its structure in three dimensions percolates, or we may say that it percolates if there exists a cross section perpendicular to the flow which percloates in two dimensions. Since percolation of a cross section implies percolation of the structure but not vice versa, these two definitions of percolation due to suspension in membranes are not the same.

Choosing the percolation of sections as a criterion implies that we consider the superficial velocity of the flow whereas choosing the percolation in three dimensios as the criterion means that we focus on its interstitial velocity instead.

The algorithms for the study of percolation by tiny particles due to attrition in membranes which I propose are Algorithm's 6.19 and 6.20. Algorithm 6.19 prepares the structure while 6.20 does the percolation simulation. Here both VT and $\mathcal{V}$ means the Voronoi tessellation. The appeal factor is the probability that a particle will choose to leave a cell via a certain bond. It is weight by the gradient of each bond, and is calculated over all bonds going in the downward direction from the cell. Transfer grids are square grids which help map the continuous plane at the top layer to bonds connected to it, that is to say, it maps a continuous Euclidean plane into discrete grids and from there on to bonds. In other words, $E^2 \to D^2 \to \{b\}$.

Algorithm 6.19 *Percolation by tiny particles due to attrition in membranes.*

generate a Voronoi tessellation in three dimensions;
transform the VT into a centroid VT;
find the cross section of its top layer;
$C \leftarrow C^2(\mathcal{V})$;
find transfer grids of C;
for every cell in VT **do**
  find the maximum chamber capacity of its cell;
  find appeal factors for all its bonds;
endfor □

Let the gradient of each bond be represented by an angle α that it makes with the horizontal plane. Then the gradient can be calculated from the coordinates of the two end points of each bond, providing that $z_2 > z_1$, from $\alpha = \tan^{-1}(z_{12}/((\Delta x)^2 + (\Delta y)^2)^{1/2})$, where the slope is downwards from p_2 to p_1, and as usual $z_{12} = z_2 - z_1$. Algorithm 6.20 describes the membrane percolation simulation proposed.

Algorithm 6.20 *Percolation by tiny particles due to attrition in membranes, percolation simulation.*

```
for each time step do
  for all arriving particles do
    find their random arrival position;
    round these positions to the precision of the grids;
    map positions on to bond numbers, using the grids;
  endfor
  for all particles do
    update distance travelled;
    update chamber crowding;
    find percolation of blocked chambers;
    if chambers percolate then
      terminate the simulation;
    endif
  endfor
endfor
```

□

Here I concentrate on the interstitial flow velocity, therefore the percolation is supposed to occur when the chambers percolate in three dimensions. Coincidentally, this also makes the calculation easier. If we were to choose the percolation of sections as the deciding factor for percolation of the membrane, for instance, we would have needed to consider approximately $2n$ sections in total, where n is the number of chambers. With an equal probability for success for all the homogeneous sections, this would still leave us on average n sections to consider before we know that a sample percolates, if it does, though we would still need to test all the $2n$ sections in cases where it does not. This number $2n$ arises from the fact that to completely cover all combinations of grouping cells into sections we need to consider for each cell two sections for each existing cell, one touching its top while the other touches its bottom.

Sphere packing is a rich field of its own, within which the packing density is generally refered to as η, an efficiency, instead of the usual density symbol ρ. The rigid packings of spheres vary in density from the lowest in loose packing where $\eta \approx 0.06$ to the highest, which is shared by the cubic and the hexagonal closest packings, $\eta \approx 0.74$. A rigid packing is a packing in which all spheres touch at least four others, and the points by which each sphere touches its neighbours can neither be all in the same hemisphere nor all on an equator, *i.e.* a greatest circular section. So we can now limit the value of ρ that we shall use to be in accord with $0.06 < \eta < 0.74$. In this early stage I shall not use a Monte Carlo study to find the probable ρ, *i.e.* η, but will aproximate it to be some value within the range mentioned. Since the most familiar sphere packing in human history must be that by which oranges are stacked at markets, expecially open markets like the one at Bolton, which gives the efficiency of packing $\eta \approx 0.74$, we shall assume that this is the way the clusters arrange themselves.

Notice also that piling oranges in a neat tetrahedral shape on a table and a packing them into a rectanbular box both produce the same crystal structure, that is the face-centred lattice, the difference being only in their habits.

For all intents and purposes the percolation probability of spheres under the influence of the van der Waals force must be the same as p_c of a face-centred lattice. This is because the biggest cluster of both cases will have the same structure and their orientation will determine the orientation of the structure. We can do away with the orientation of other minor clusters precisely because they are much smaller, which justifies our grossing over their individual shapes and only concern ourselves about their statistics, that is to say, their number. I think that stacking oranges into a box is cubic close packing while a pile of oranges is hexagonal close packing, but this needs to be checked.

To find p_c of the close-packed cluster of spheres, one needs a program similar to the one mentioned in § 6.10, but which would do the job for three dimensions instead of two. For this purpose, the program for 2-d tilings mentioned in § 6.10 has been developed further to deal with

regular lattices in three dimensions. At first I thought that there should be some other way to do this instead of having to develop another program for a general lattice in three dimensions, since this is already the last week of the project and time is running out. But in the end I found it far better to spend some time to systematically develop a program for general cases than to opt for some adhoc approaches. As a result, a program that creates regular lattices in three dimensions for the purpose of percolation study has been written and is listed in § A.28.

As the 2-d program in § A.6 does for all 2-d regular lattices, this new program can deal with all possible lattices in three dimensions. The difficulty is, however, in the meticulous nature of identifying all the vertices and links in each unit cell. In this respect, the cubic close packing is much simpler to do than the hexagonal close packing. Therefore I will only do the first one while leaving out the second, which ideally could be used for the purpose of comparison.

Figure 6.10 shows the lattice generated by the program and one which is used for finding the percolation threshold.

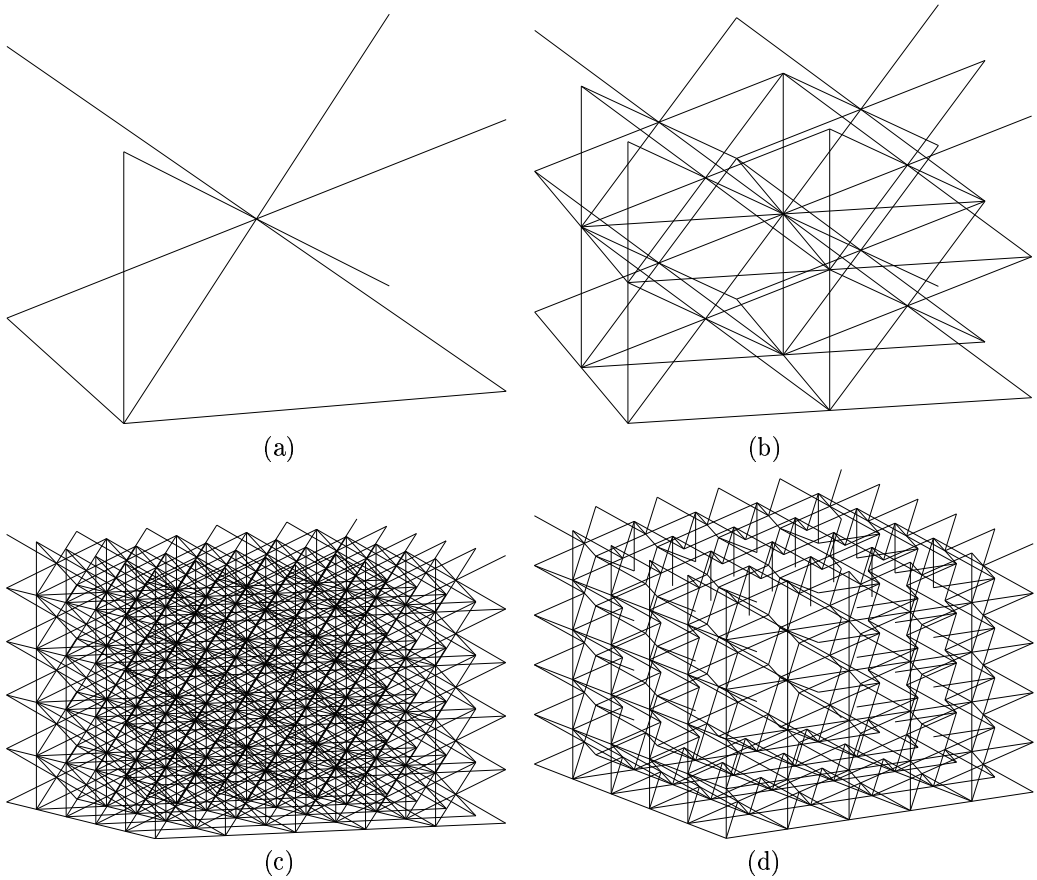


Figure 6.10 *Percolation of the cubic close-packed lattice;* (a) *a unit cell,* (b) *eight unit cells, one from each of the eight groups,* (c) *network of size* $5 \times 5 \times 5$ *unit cells, which is used in a simulation, and* (d) *the same network with only boundary edges drawn to make it easier to look at.*

At present the program only finds p_v, p_e, x_v and x_e, not p_c, p_b, x_c and x_b. There may be altogether three types of cell and bond pairs, in comparison with two in the 2-d case, depending on whether the number of shared vertices required be 3, 2, or 1. For our purpose in the study of filtration, we only need to know p_v, which, when generated from a $5 \times 5 \times 5$ network as shown in Figure 6.10 (c), turns out to be 0.25. All the results from simulations are $p_v = 0.2501 \pm 0.0400$, $x_v = 8.0645$, $p_e = 0.1320 \pm 0.0209$, $x_e = 17.1709$, while $n_v = 341$ and $n_e = 1{,}375$.

This means that when the space will be blocked, *i.e.* percolates, when it is filled up to one quarter of its volume by suspended particles in the form of clusters of the highest packing density. Because the cubic close-packed spheres fill 0.74 of the space, this ratio translates into the real volume ratio of $0.74 \times 0.25 = 0.185$, that is 18.5 per cent by volume. If the fluid in our system is water, then $\rho = 1{,}000$ kg $\cdot$ m^{-3} and the percentage by volume above is equivalent to a density of of the suspended particles of 555 kg $\cdot$ m^{-3}. Notice also in our simulation that the cluster shapes need not be convex.

In the light of the symmetry between particles and space, in other words between particles and anti-particles, which give rise to a symmetry and the three types operational regions that I originally proposed in a study of traffic congestion (*cf* § 7), the operational space of our filter may fall into three distinct regions when it is subjected to very small particles suspended in a fluid.

If we specify by ρ_v the ratio of the volulme occupied by all the clusters to the total volume, and ρ the density in weight per volume, and if the three regions of operation are labelled I, II and III, then in the case of I, $0 \leq \rho_v < 0.25$, while for II, $0.25 \leq \rho_v < 0.75$ and for III, $0.75 \leq \rho_v \leq 1$. In other words, for I, II and III, we have respectively $0 \leq \rho < 555$, $555 \leq \rho < 1,665$ and $1,665 \leq \rho \leq 2,220$, where the unit of ρ is kilogram per cubic metre. Generalising this, the regions I, II and III correspond respectively to $0 \leq \rho < \rho_{c_1} = \rho_1$, $\rho_1 \leq \rho < \rho_{c_2} = \rho_2$ and $\rho_2 \leq \rho < \rho_c$, where ρ_c is determined by a physical constraint, namely the packing efficiency mentioned.

Qualitatively speaking, these three regions may correspond to the operational-, blocked and non-operational regions. If our system also contains other particles which are larger than the pores, then in Region I filters will operate normally until the effects of blocking or blinding by the large particles become prominent, as has been studied in literature (*cf* Jackson, 1994; Jafferali, 1995). In this case fouling of the filter is caused exclusively by the blocking or blinding of these larger particles, which result in the formation of cake, and unless $\rho = 0$ there will be some blockages of internal pores due to the blockage caused by small suspended particles forming cluster. This latter type of blockage, which is of our concern, is to be expected due to various reasons. Cluster formation may be caused by nonhomogeneity in the concentration of the suspension which raises the concentration in some region such that it exceeds ρ_1.

Additional reduction in the flux is to be expected from two reasons. Firstly blocking clusters may block some of the pores. And secondly, the suspended particles together with free clusters, *i.e.* those which are smaller than they could block pores, displace the volume of the liquid surrounding them and thereby reduce the flux. The first one of these will produce an effect similar to blinding described in literature (*cf* Jackson, 1994; Jafferali, 1995, *ibid.*), where no backflushing may recover the filters to their virginal state. On the other hand, the majority of those particles and clusters in the second scenario is expected to be easily removed when backflushed. Among these latter there could yet be some which adhere themselves to the walls, whose fixation defies backflushing. But these last ones are expected to be small in number, and thus can be neglected, because we shall assume that the combination of ρ and channelling results in the possiblity that pores are blocked being very close to either zero or one.

The channelling effect, or the occurence of rivulets by some other authors, is the formation of preferred paths through a porous media through which liquid and the solids it contains pass. It is not yet clear what these rivulets would do to our system. If all the channels channel equally both the liquid and the solids, then the blockage along these path may be expected to rise above the average value of the whole structure if only because this becomes more probable statistically. But if there exist some channels which prefer channelling liquid to particles or vice versa, then the effect they produce will vary and become complicated. For example, channels which like to channel particles are more likely to find themselves blocked in the end by those particles which pass through them. On the other hand those channels which channel liquid better than solids will be less prone to blocking on average, but will leave other pores around them with the excess particles, and these latter will necessarily become blocked more often than usual. But, for our purpose here, we shall assume that it is solid particles that are being channeled. This should raise the possibility of blocking in some of the pores by certain amount. Channelling in general needs further investigation which will not be covered here.

Before going on to the next step of our study I should briefly mention the core idea that makes the program in § A.28. There are eight types of unit blocks here, compared with the four types in the case of the program in § 6.10. These correspond to the area drawn and labelled in Figure 6.11.

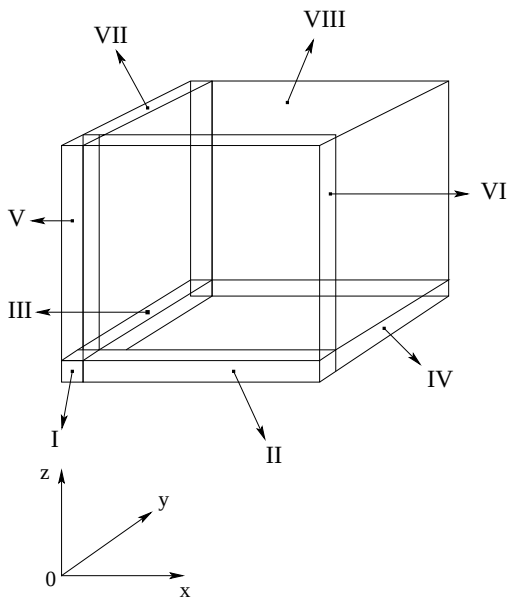


Figure 6.11 shows the eight areas defined by the eight types of unit blocks they contain. Area's from I to IV correspond to those previously defined for the 2-d program. Although unit blocks in the various areas works differently, that is to say, they adopts different set of vertices from differnt sources, and create different edges, all of them follow the same four rules. These four rules in the mnemonic not cryptic forms which I use are, *take vertices from behind*, *make front vertices*, *draw edges behind* and *draw no front edges*. With these rules in mind, both programs should become self-illuminating to such extent that no further explanation is needed. This set of rules does two things, namely organise vertices and then link them with bonds. The unit vectors in the three directions being orthogonal means that we will have a nice and square end product suitable for a percolation study.

Figure 6.11 The eight areas defined by the eight types of unit blocks.

In fact it is wrong to say that all four rules work differently for each unit group. Only the two on vertices, *viz.* the first two, are different. The rest, *viz.* the last two which concern edges, are the same for all basic units. These are the only two crucial tasks with the discovery of which the writing of both program becomes worthwhile.

The input data needs only contain details regarding units in Area II, III and V. Area I, being at the origin, is trivial, or should one rather say unique. Area IV can be derived from Area's II and III. Like wise VI is derived from II and V, and VII from III and V. Finally Area VIII turns out to be nothing but II, III and V combined.

So far we have only mentioned the situation where $0 \leq \rho < \rho_1$. In the second case, where $\rho_1 \leq \rho < \rho_2$, many more pores are blocked from small particles than in the first case. The concentration is already beyond the first critical point. But while the second critical point is still not reached, there would still be an infinite cluster of space – in this case the liquid – surrounding the particles. In other words the space still percolates. The presence of this infinite cluster, or continuum of the medium, means that the filter can still be in operation until it should be blocked or blinded by larger particles which individually can physically block the pores as found in existing literature earlier mentioned.

The third and last case, where $\rho_2 \leq \rho \leq \rho_c$, represents the extreme which can be easily comprehended. Here the solid particles occupy more space than the liquid does, as a result of which the combination is no longer a suspension but a slurry. Clay material produced by this slurry would block most of the pores within the structure and make filtration impossible. Backflushing will not be effective on filters which have undergone such fate.

§ 6.11 The second part, flow through the cells

Next we will investigate briefly the effect that channelling may have on the value of p_c. There are three cases considered here. The program for this purpose is listed in § A.29. The program creates a centroidal Voronoi tessellation in three dimensions. The first set of simulations works on a normal case of cell percolation, similar to that in § 6.11 but here the system is a centroidal Voronoi, which implies a constraint on cell sizes and distribution. The function `perd` in it is obtained by inputting the `Blocked` variable instead of generating it internally.

In this section, I write as I go along with my developping the programs, so the contents should be easier to follow than in other sections. This is not to say that in those other sections I had not kept records of what I discovered. Every Ph.D. student starts off doing his project knowing that he should write along as he goes, and plans to nothing but that. But the truth is that eventhough we always write, but the way we write develops with our experience. Also, with the the increase in the understanding of our problem, we no doubt would be able to write a better description of what we do and how we do it. I think that this is an unavoidable fact, and feel that it probably is the reason why we should keep on working.

So much for an aside. From one hundred generators originally, Voronoi operator is applied twice. After the rims has been trimmed there are 280 centroidal Voronoi cells remaining, and this is the value of n_c. For x_c the value is 10.7929, while p_c from 2×5 simulations is 0.2314 ± 0.0602.

Next investigate the effect of channelling by assuming that the steepest gradient of all the bonds arriving at a cell decides how quick it percolates. Here cells are sorted according to their steepest gradient of incoming bonds. Working on the same network as previously, if the percolating order is such that the steeper the quicker, then $p_c = 0.2107$. But if on the other hand steeper incoming bond means slower percolation, then $p_c = 0.1429$. The critical probability is constant in this case since the order of percolation is predetermined by the orientation of bonds with respect to cells.

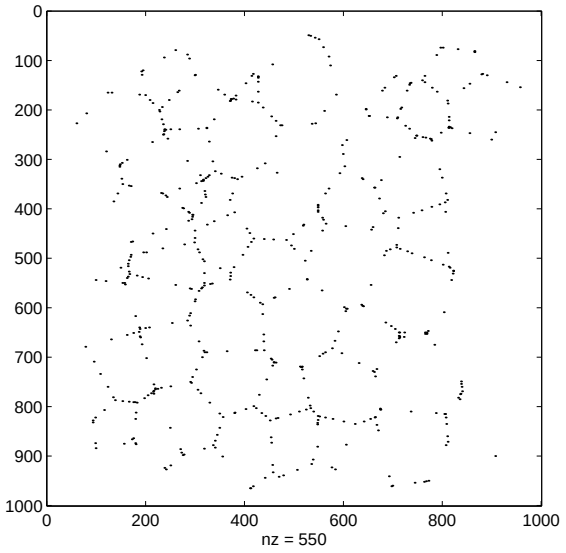
If instead of looking at only a single bond we take the signed summation of bonds entering and leaving a cell, then $p_c = 0.1321$ when the criterion is $\min(\sum b_i - \sum b_o)$, and $p_c = 0.1393$ when it is $\max(\sum b_i - \sum b_o)$, where b_i and b_o are respectively the incoming and outgoing bonds.

Our studies up to now tell us that if the suspension is homogeneous and there is a rivuletting effect, then the location of the blockages made by clusters of suspended particles among all pores of the structure is predetermined. This is in contrast with the blinding and blocking of large particles, where such location is random. In fact, even for these latter large particles, the location can only be random when the particles have a variety of sizes. It can never be wholly random, however, if this is not the case, since in the former case the randomness is introduced by the distribution of sizes which is random, but in the latter the blinding or blocking will be determined by the size of pore openings which is fixed by the geometry of each network. The randomness then can only be in the order not location of blockings.

The next study is a combination between continuum and network percolations. Here suspended particles are grouped into quanta, each channelling through a path or rivulet of interstitial distance with some interstitial velocity.

At the top, the layer of the structure where the incoming particles arrive, is a cross section all the cells of which are gridded to provide means to determine the path at the beginning of each quantum. The partitions in this layer is conveniently found by cutting some faces of the convex hull of each cell in the top layer by the plane $z = 0.9z_{\max}$, where $z_{\max}$ is the maximum z-coordinate of all the cells under consideration.

Given coordinates of two points, (x_1, y_1, z_1) and (x_2, y_2, z_2), and a plane equation $z - a = 0$, we may think of the plane equation as being one coordinate given, $z = a$, and find the coordinates of intersection between a line passing through the two points and the plane from the parametric equations for the line. Parametric equations are in fact interpolation done on each of the coordinates. In this case, which is useful when finding the intersection between an edge of a triangle and a plane perpendicular to some coordinate axis, the parametric equations are $x = x_1 + x_{12}t$, $y = y_1 + y_{12}t$ and $z = z_1 + z_{12}t$. From the plane equation $z = a$, therefore $t = (a - z_1)/z_{12} = (a - z_1)/(z_2 - z_1)$.



The program being written finds the intersection of cells with the horizontal plane by finding the intersection of the faces of its convex hull with the same. Since every face of the convex hull is a triangle, the program essentially finds intersection between edges of these triangles and the horizontal plane. The result obtained from an intermediate state during the course of development of the program is shown in Figure 6.12. The partitions look incomplete because the picture is taken as a test while developping the program as mentioned. A picture with the same degree of incompleteness as this one is not to be obtainable from the comleted program.

Figure 6.12 Intersection of convex hull faces and the horizontal plane.

For each face of the convex hull that intersects the plane, there will be two points of intersection arising from the two edges of the triangle intersecting it. Let (x_1, y_1) and (x_2, y_2) represent these two points. Then we may scan up in the y direction finding $x = x_1 + (y - y_1)x_{12}/y_{12}$ for each y along the way, and then scan in the x direction, this time finding instead $y = y_1 + (x - x_1)y_{12}/x_{12}$. Aferwards we could fill in the space by scanning along the x direction for all y positions.

Figure 6.13 shows the progress of my programming the codes, step by step, trying to close all the partitions such that no gaps remain. Two problems have been discovered, namely those of rounding and precision. Before the correction the resul looks like Figure 6.13 (a) and (b), and after rounding problem corrected Figure 6.13 (c).

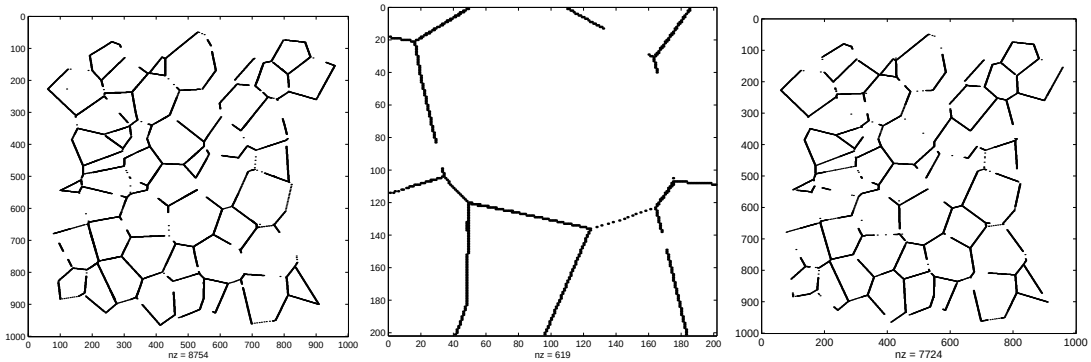
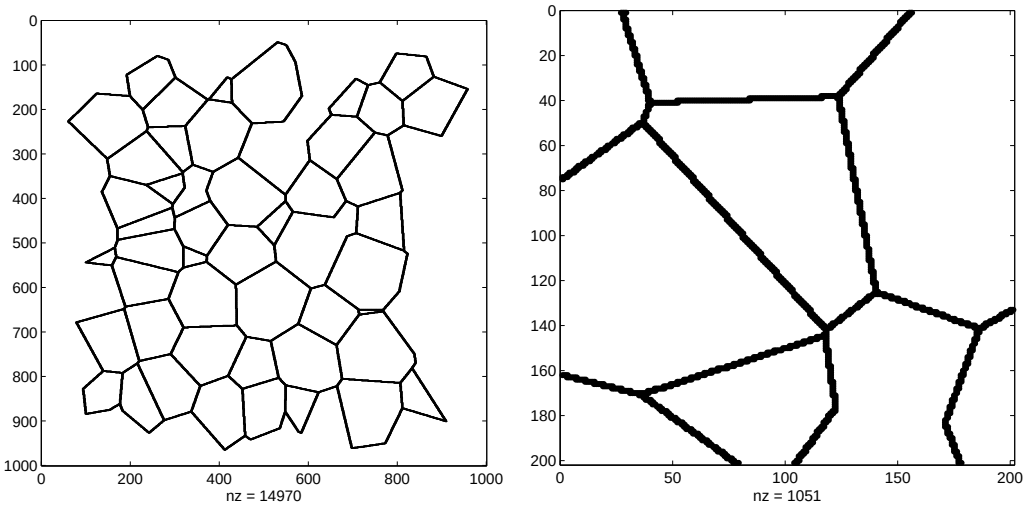


Figure 6.13 *Correcting the effect of rounding, in other word discretisation.* (a) *The gaps resulted from rounding or discretisation,* (b) *a closed-up view of (a) and* (c) *partial remedy where roundings have been solved but with the degree of precision not yet raised.*

After having corrected the problem regarding precision, by increasing the number of steps when calculating x or y, the result still misses several walls, the cause of which is still unknown at present. This is shown in Figure 6.14.



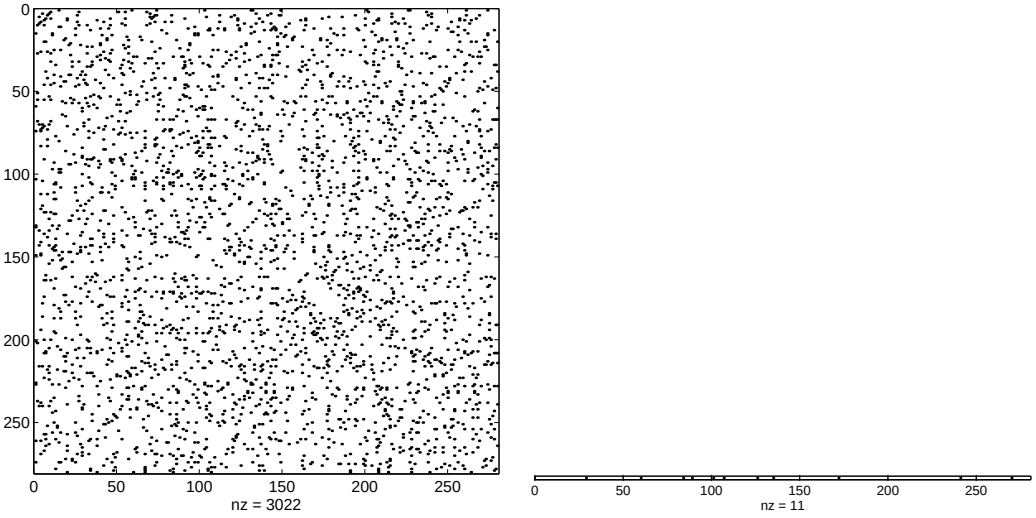
(a) (b)

Figure 6.14 *After having increased the precision to ten times the previous value,* (a) *and a closed-up view,* (b).

At first I thought that there might be some triangular faces missing from the surface of the convex hull, which would have accounted for the missing boundary in the section. But after having tested the minimum number that an edge of each hull appears as the edges of all its triangular faces, and see that it's value is correctly two, this becomes out of question.

The figures, *viz.* Figure 6.12, 6.14 and 6.14, are produced from the `spy` command in Matlab, as a result of which the x-axis runs downwards while the y-axis runs to the right. This command looks at a matrix from above as we look at a map. In the present case our matrix is a full-, not sparse matrix. The number written at the bottom is the number of all its nonzero components, which is less than the number of times that we calculate them since we need to calculate some of the points more than once in order to increase the precision to eliminate gaps in other places.

The command `spy` is used more often with sparse matrices since these are often too large to list, and listing their members in pairs makes it difficult to visualise. As an example, Figure 6.15 (a) is what we get when we `spy` our neighbour matrix `necc`, while in Figure ℵ (b) are all the neighbours that the 100^{th} cell has. Cells which have few neighbours generally live along the border. For example the 110^{th} cell has only three neighbours, and it is located not far from the lower x limit.



(a) (b)

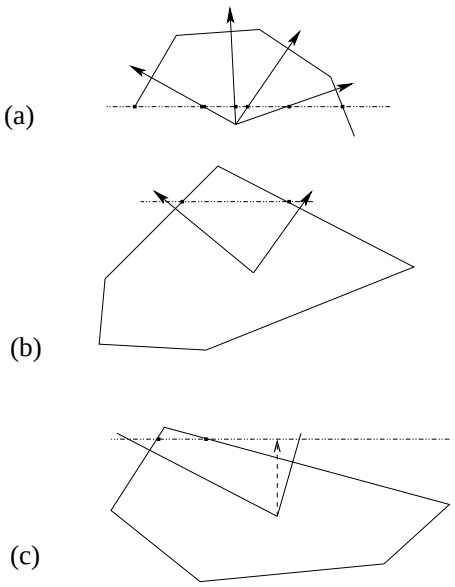
Figure 6.15 *The result when we* `spy` *our neighbour matrix. Here* (a) *shows the neighbours of every cells while* (b) *only shows those of the* 100^{th} *cell.*

At this stage I recheck the neighbour matrix of the program, and find that it includes poorly defined neighbours, that is those which only have one vertex in common. Therefore I changed the program first to only look for neighbours who share at least two vertices. But this has shown no

noticeable changes in the results that we have so far.

The next step is to colour the cells. Like a child painting a picture, there are so many ways one can paint or label the tiles of a tessellation. For example one could draw a vertical line first and then branch out either horizontally or diagonally. On the other hand, one could also draw the diagonals first Yet another way is to expand radially, spiralling outwards. With parallel computing we could also divide the area into domains, paint each domain, and then merge the resulted areas together.

But here I opt for painting the cells by scanning horizontally, moving upwards in layers. Once past a wall, the program moves on until the next wall is reached while gathering all the grids that are between the two walls into one group. It then colour the whole group by a colour picked up from one layer below it. If no colours exist, then it creates a new colour which in turn gradually propagates upwards this way until the upper wall is reached.



The intersection of bonds will not always coincide with the intersection of the cell, neither is the projection of a cell perpendicular to the plane the sectional plane of the cell. This is because of the three possible situations shown in Figure 6.16. In Figure 6.16 (a) the cell section contains several points, while both (b) and (c) contain none.

Figure 6.16 Sections of bonds and cells.

And here sadly the time runs out, so I will suffice myself to describing what I see should be done next. Up to now we have a three-dimensional network and its top section. We also have the list of all its bonds, which contains the connections and the draining angles sorted in a descending order.

Next we should find a mapping from each cell section to the corresponding nuclei. Then the times it takes to traverse each bond must be calculated. This time for each bond is then divided by half, one belonging to each of the two cells connected by the bond.

When it comes to bombarding the filter with our small particles, we can not keep track of millions of particles and therefore we should quantise them into units. These units or quanta can then be treated as individual particles. When a quantum enters a cell, it is assigned t, the time to reach the nucleus. Later time sees this t decreases in steps until it finally reaches its destination, the nucleus. Once there, it is assigned the next bond to go along, taking into account what bonds are available at the time and their comparative probabilities, which in turn depend on their gradient as mentioned. When this is decided, it is given t_1, the time it would take to reach the border that lies at mid point of the bond.

This goes on forever, apart from that at each time step we look to see whether the blockages in our filter has percolated. After updating the list of blocked cells, if we find that percolation has occured then the simulation would end. Percolation occurs in each cell whenever its concentration has reached a certain value. This value we have found in § 6.11 to be 18.5 per cent by volume.

To calculate the flux decrease we find instead the decrease in the superficial area. This is calculated from the total volume of the void subtracted by the volume of all cells that had percolated, and then subtracted by the total volume of solid particles which are suspended inside the network. The area of the cross section is then the volume which remains divided by the thickness of the filter.

It is not a little to have to leave things unfinished after having started it off. But as one New Zealander poet says, 'Alone we are born and die alone, yet see the red-gold cirrus over the snow mountains shines. Upon the up-land road ride easy, stranger. Surrender to the sky your heart of anger.' And with this I go on to the next chapter.

7

§ 7. Percolation in traffic modelling

It does not take much imagination for an average person to see that traffic congestion is a percolative process. As I have lived and studied in Bangkok where, at that time, the traffic jams were renowned as being second to nowhere in the world it would have been peculiar indeed if after having acquainted myself with the percolation theory I do not apply it to the study of traffic systems. I remember once in 1990 it took me more than five hours on a bus to travel the distance of ten kilometers or so along the Sukhumvit Road in Bangkok. Being then in my last year at the university and having missed an important meeting with my supervisor that day as a result, the reason I gave to my supervisor must have sounded like a lie. Only the newspaper of the following day could save my integrity. In it there was a report that the road on which I had spent a whole day travelling had been given a footage on the BBC World news.

Soon after having decided to investigate the application of percolation to traffic, I gave a poster presentation at the Fourth Annual Conference of Thai Researchers in Japan in 1997, the proceedings of the event of which also contain a number of abstracts from some of the researches that I was currently working on (Tiyapan, 1997, KNT2(ii)). The poster presentation was under the title *Fractals in the modelling of traffic networks* and was reasonably a success in that it interested several people there and I was asked a number of interesting questions. One of the two honourary chairs of the event suggested that I wrote a detailed article for the monthly journal Sakkayaphab (*Śakāyābhaḅ*) which is published by the Association of Thai Professional in Japan (ATPIJ), but such a paper never materialised as I found it difficult to write about the topic in Thai and I was reluctant to have my article translated from English again because there have been mistakes in earlier such attempts and the editing team does not always bother to tell the reader that the articles are translation (*cf* Tiyapan, 1996, KNT2(ii)). One mistake which stands out is the translation of a passage in my manuscript,'The irony in this is that while the money involved increases exponentially in relation to the cases of its counterpart which are known as the *blue-collar* crimes, the risk of being caught is comparatively small', into a translated version which means the opposite, 'Naịkhảṇadị ạjàkamr [sic]‡ ịk prabhẹd gụe *Blue-collar crimes␣ zù̀eng bọaban kaḅ ngoen caṃnoan mạk. Mù̀a prịaḅdịaḅ kan lā̀ew ạjàkaur* [sic]¶ *lhăonī̀ thụk caḅkum mạk kvằ ḅạeḅ rạek.' [While another type of crimes are Blue-collar crimes which involves a lot of money. Comparatively these crimes are caught in greater number than the first type.* Whether I could write good Thai or not, from this lesson I think that from now on I should translate my own articles.

In fact, it was not so much a poor translation of my article that I mind but the failure of the editor to acknowledge that the article was a translation. As it is the case, whether this be coincidence or not, translation quality tends to be better when the name of the translator is prominently written (*cf* Tiyapan, 1996, KNT2(ii)). In the reference above, only minor mistakes appear in the translations, namely 'however' translated into a word which means 'moreover' (*yang*) and similarly 'metaphor' into that which means 'simile' (*upàmạ*).

The paper in § E.20 is, except from a few minor spelling corrections and the changed format of the bibliography, an exact reproduction of the paper I submitted electronically to an international journal based in America, either the Journal of Statistical Physics or the American Journal of Physics I am no longer certain since I have no access to the backup copies of my own files on the computer in Japan.

I feel that these minor changes are justified if only for the reason that the paper has not been accepted for publication. I include it here in order to show the development of ideas leading up to the present work. Towards that end all the contents, and as far as possible even all the syntactic and grammatical mistakes, have been left untouched. Some of the figures have been left out due to the loss of my home directory on a computer in Japan which I have learnt about soon after my arrival to resume my Ph.D. research in Manchester. The address of correspondence listed is my

---

‡ should be *ajàyạkamr*

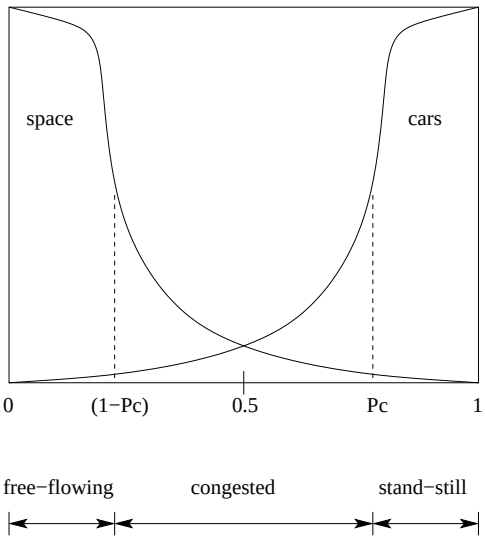
¶ should be *ajàyạkaur*

home address eventhough I was then with the Tokyo Institute of Technology. This was only to avoid further misunderstanding in cases where letters to me had been opened before they reach my hand without due apology. I suppose that the idea about the unity of a laboratory there amounts to a different standard of privacy and respect from what I have been brought up to believe in. There, for example, the circulation of preprints received for refereeing among the nearly twenty members of the lab is a matter of course.

The supposedly inferior quality or contrivedness of the paper can probably be ascribed to various factors, for example the constant dread of always being observed (though I have no doubt whatsoever about the relevancy of such topic to modelling in control systems), constant and to me somewhat purposeless drive towards publications by all possible means, and last but not least my own shortcomings. Nevertheless, I still believe that the approach I proposed there is sound and the idea new.

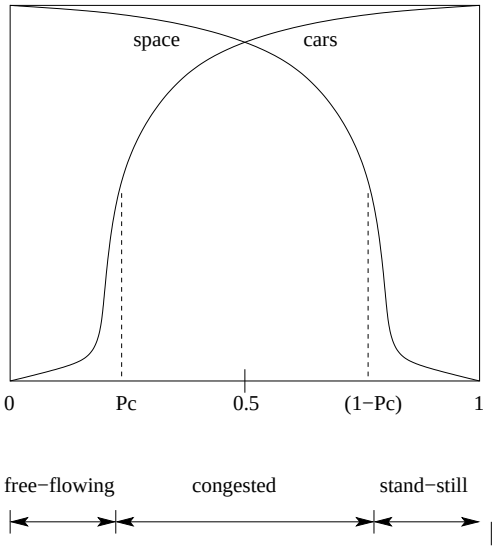
My interest in traffic began at the age of thirteen when I as a boy scout volunteered to become a traffic scout to help managing the traffic, mainly around my school but also occasionally at some other places around the city of Chiangmai, Thailand. The work, which I had done for two years, has given me a first hand experience but is not directly related to the following modelling of traffical network.

In the simulation the networks are drawn which have their vertices as points where two or more roads meet one another. The proposed study is to compare the robustness of two traffical networks by comparing their percolation thresholds. When a new motorway is planned, for example a ring road around a city, the two networks, one with the ring road and the other one without, can be simulated to find their percolation thresholds and then these values compared.



Traffic status or condition when the percolation probability p_c of cars is more than 0.5. The traffic condition is generally described as free-flowing, congested, or stand-still. For cases where $p_c > 0.5$ as the one shown in Figure 7.1 the stand-still traffic corresponds to the situation where only cars have percolated but not the space, that is the places on the road available and accessible to the cars. A free-flowing traffic is where only the space but not the cars has percolated, and a congested traffic is that when neither of the two has.

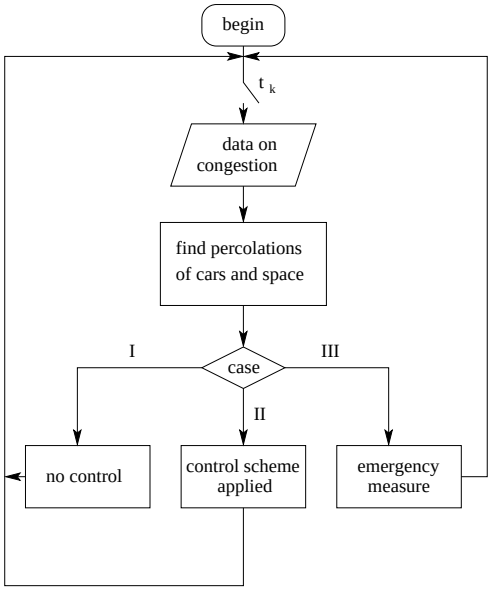
Figure 7.1 Traffic status, $p_c > 0.5$.



Traffic status or condition when p_c is less than 0.5. On the other hand, in the case where $p_c < 0.5$ shown in Figure 7.2 the critical probability of cars is now on the left hand side of the middle line instead of on the right hand side thereof in the previous case where $p_c > 0.5$. Definitions of the free-flowing and the stand-still statuses remain the same, but the congested traffic is now the traffic where both cars and space have percolated. An interesting question is whether or not there is a difference between the congested area in Figure 7.1 and the one in Figure 7.2.

Figure 7.2 Traffic status, $p_c < 0.5$.

A *modelling algorithm* which studies the percolation of a traffical networks starts by finding all the vertices and edges forming a network from the road data. Then a blocking algorithm operates by randomly shutting off one edge after another until the network percolates when the critical probability of the network may be calculated. A *control algorithm* for the real time traffic control, however, is as shown in Figure 7.4.



Let C means *cars have percolated* and S means *space has percolated*, then in Figure 7.4 the cases I, II, and III are respectively $\neg C \wedge S$, $(\neg C \wedge \neg S) \vee (C \wedge S)$, and $(C \wedge \neg S)$. Case I is the normal congestion, nothing to worry about. Examples of the control schemes used in Case II are overriding of the traffic lights by the manual control of traffic polices at certain strategic points, temporary one-way systems, bird-eye view observation and feedback from helicopters, traffic control centre, distributed control centres together with traffic radio channel broadcast. And lastly the most important and critical Case III the emergency plan of which may include directing all cars away from congested clusters or if necessary out from the city, and directing all incoming traffic such that no more cars may enter the city until the emergency status ends.

Figure 7.4 Proposed traffic control in real time..

A congested cluster can be broken up by forming a one way flow channel cutting through it which leads cars away from the cluster. How far the channel needs to go before letting the cars on it circle and seep back into town depends on how badly congested the traffic is. Although it is a normal practice to lead cars along a long detour because this brings in more road surface and thus enlarges the network, identifying the percolationg cluster and cutting it into two or more parts by guarded flow channels is much less common or even unheard of. I think the latter is more important and will lead to a better and more effective control, namely the control fo the percolating, in other words the biggest, cluster.

The percolation probability is important for the networks of traffic both inside cities and among them. It shows the degree of connectivity of the area being considered. Urban road networks have a general character which differ from one country to another, the simplest construction of which seems to be that of the square lattice. One can find this theme of the square lattice and its variations, similar to the dislocations and defects found in minerals, in America. Examples are Denver, Aspen, Durango, Pueblo, Salida and La Junta in Colorado; Boise, Pocatello and Twin Falls in Idaho; Butte, Bozeman, Coeur d'Alene, Kalispell in Montana; and Cheyenne, Laramie and Sheridan in Wyoming (*cf* Florence *et al*). The Great Junction in Colorado and the Great Falls in Montana are very close to being perfect square lattices. As more examples of these (*cf* Collins USA, 1999), in Arizona there are Phoenix, Yuma, Tucson; in California Bakersfield, Central San Francisco, Central Sandiego, Fresno, (Central) Los Angeles and vicinity, Modesto, Sacramento; in Colorado Fort Collins, Denver and vicinity, Greeley; in Florida Central Miami; in Georgia Central Atlanta; in Illinois Champaign and Urbana, (Central) Chicago and vicinity,, Quad Cities, Rockford; in Indiana Fort Wayne and Indianapolis; in Kansas Topeka and Wichita; in Louisiana Central New Orleans; in Maryland Central Baltimore; in Minnesota Central Minneapolis and Central St. Paul; in Missouri Central Kansas City; in Nebraska Lincoln; in Nevada Las Vegas; in New York Manhattan; in Oklahoma Lawton, Norman, Oklahoma City and Tulsa; in Pennsylvania Central Philadelphia; in South Dakota Sioux Falls; in Amarillo, Central Houston and Lubbock; in Utah Central Salt Lake City; in Washington Central Washington D. C. and Central Seattle; and in Wisconsin there is Central Milwaukee. One example in Canada is Toronto. The square lattices of these cities are sometimes cut through by motorways or interstate highways as is the case in Amarillo, Texas. Or they can be surrounded by a ring road or a county highway as is what happens with Lubock, also in Texas.

When the percolation probability is greater than 0.5, we have the interval $p_c \pm (p_c - 0.5)$ where neither the blocked nor the free roads percolate. If this interval is narrow, that is if $p_c - 0.5$ is small, then within this interval the condition of the traffic is very sensitive, and even a seemingly small change may lead to a standstill or instead to a free-flowing traffic. This is easily visualised, since in such situation there would be small islands of free-flowing roads within a large congested cluster, and vice versa small clusters of congested roads within an otherwise noncongested area. Despite their sizes, such small islands of anomaly in either of the phases are particularly important. Moreover, their importance increases the closer p_c is to 0.5. The same characteristic happens in politics where, in the case of coalition governments, a minority party which has relatively few representatives can become critically important and influential to the major party when the latter needs them in order to be able to govern (Ireland, 2002).

Plan to build a ring road around a citie usually includes flyovers, overbridges, and tunnels in order to avoid having intersections. The design philosophy used in Europe is to have heavier traffic goes under a lighter one, which results in either the ring road going into a tunnel or having overbridges or viaducts over it. The philosophy used in Thailand which used to be for the heavier traffic go over the lighter traffic, which puts a more severe limit on the weight of trucks and is therefore not economic in the long run, but this has started to changed, if only to follow the practice of the west. Ring roads do not necessarily resemble a circle, as in the case of the circular speedway proposed around Saint Petersburg (Petersburg, 2001), which is in the shape of a cashew nut. It will link the arterial roads of the city to Helsinki, Kiev, Moscow, Murmansk, and Tallinn into a network. The route proposed is 155 kilometres in length, has 31 bridges, 16 overbridges, 55 viaducts, and will support the volume of traffic of 21 million tonnes. The implementation and contract work is looked after by a joint-stock company KAD Sankt-Peterburga, under the order of St. Petersburg and Leningradskaya Oblast.

One classical example of a ring road is the M25 motorway which forms a circumscribed ring around London. Another more recent example is the M60 orbital motorway around Manchester. In the case of M60, various sections of existing motorways have been put together and renumbered.

The northwest quarter used to be M62, the southwest one M63, and parts of the remaining used to be M66. The motorway forms a complete ring around Manchester since 2000 with the opening of the final northeastern part which stretches from Denton to Prestwich. There is another smaller ring, an inner ring, which is formed by the A6010 and A576. Orbital motorways around cities have now become indispensable and are the hallmark of a city.

§ 7.1 Traffic networks

In his paper submitted to the Journal of Statistical Physics Tiyapan (1997) introduces a new idea of considering the development of clusters in both phases at the same time. Applied to the context of traffic network, these phases are namely cars and spaces. Furthermore, because both phases reside in one and the same network, there is a symmetry which divides the probability space into three regions, symmetric with respect to $p = 0.5$. This helps divide the traffic condition into three regions as existing literature in traffic study at the time described, namely free flowing, congested and stand still. In particular, this idea explains the difference between the congested and the stand still states. There has been no reply from the journal.

However, the author of this unfortunate paper has later come across one paper by two mathematicians in the U.S., coincidentally published in the Journal of Statistical Physics, which uses the very idea he laboriously introduced almost exactly four years ago without having received a word from either a reviewer or the editor of that very journal. Thus Gray and Griffeath (2001)‡ reintroduced the idea of anticars, which are essentially vacant spaces, antigaps and antibonds, which purportedly, greatly helps the derivation of their theories thanks to the symmetry involved. Moreover, one reference made therein (Sipress, 1999; who starts his article with, 'Why is traffic so damn bad?') compares the free flow, synchronised flow and heavy traffic respectively with gas, water and ice, and explains that as the latter's can be explained by phase transitions, so can the former's. But this is just what I have tried to explain back in 1997, that in the case of the gas the spaces had percolated but not the water molecule, in the case of the ice it was vice versa, whereas in the case of the water neither had yet percolated, or both of them had yet to percolate. When explained this way, it becomes clear that the water phase within the air as a network has $p_c < 0.5$. And so the case also must be with *antiwater*, for that matter.

According to Sipress (*ibid.*), the modelling of traffic is a hot subject in which Nobel prize winners, and cold war physicists alike, are flocking to produce world class scientific papers, which seems to include no Tiyapan. But can plagiarism be excluded?

‡ Lawrence Gray and David Griffeath. The ergodic theory of traffic jams. Journal of Statistical Physics. vol. 105, nos. 3/4, 413–452. November 2001.

§ 7.2 Percolation of road networks

I began my study of the traffic networks in 1997 while in Japan. In the same year I submitted two papers, § E.19 and § E.20, to the Journal of Statistical Physics, electronically as it was required thus. There was an automatic email which said that my papers had been successfully received. Apart from this, I heard nothing from them, and several months later, when I wrote to inquire, there was no reply. One of these two papers was on traffic congestion. In it I introduced the new idea of considering the percolation of space in accompany with the usual percolation of congested cars. If we do this, our operational space will be readily divided into three parts instead of two, which correspond with the three known traffic situations, namely free-flowing, congested and stand-still.

Recently I found out that there have been some papers published in 2001 in the Journal of Statistical Physics which used this idea which I tried to introduce back in 1997. I was totally flabbergasted, and could not decide what to do. I do not know whether I should write and complain to the editor of the journal, nor do I know whether this would be a worthwhile thing to do. I do not know whether to exclude plagiarisation.

I have developed a program and a procedure for finding the percolation thresholds of road networks. The programs are listed in § A.25. These are different from the usual percolation programs used for other kinds of networks including Voronoi. Because of the existence of flyovers and elevated express ways, we can not use the duality operator to transform a network of vertices and edges into one of cells and bonds. Having said that, the said transformation could become useful in the future in some other applications, in some other areas or even within the study of traffic network itself. But at present I only have one application in mind for the dual networks of roads, and that is related to fire prevention where such an application is by no mean obvious. Therefore, here the boundary is defined again to obtain the cells which represents the zones.

The codes also contain data of several towns and cities, namely Amsterdam, Brussel, Freiburg and Manchester. There are three main datasets. The first one contains a list of the coordinates of all the vertices. The second one is a list of edges, together with the numbers of the cells that each of them connects. The third one contains the coordinates of turns in each of the roads listed in the second dataset. This is in order for the graph to look like the actual roads it represents, instead of containing only straight lines, as would have been the case were the windings of the roads not to be taken into account. Also, these coordinates will make it possible to calculate the true length of each road. Eventhough we have no use for these lengths at the present stage, future developments may need them.

The procedures which I have developed for gathering the data and processing them can be carried out by a single researcher, and require no sophisticated tools. Were these tools become available in the future, the former could be adjusted to accommodate them.

To my surprise, so much so that I first thought that there was something wrong with the program, the mean coordination number of the road networks of Manchester turns out to be exactly 3. The second simulation gives $x_v = 3.0513$, which is still very close to three.

(a) (b)

Figure 7.5 *For Manchester in this picture, $n_v = 220$, $n_e = 330$, $n_c = 25$, $n_b = 50$, $x_v = 3.0000$, $x_e = 4.7273$, $x_c = 4.0000$ and $x_b = 6.5200$.*

Figure 7.6 shows the largest clusters plotted against p, the probability that each vertex, edge, cell or bond respectively for (a), (b), (c) or (d). Here p_v^s is the percolation probability of the space in the network of vertices, and similarly for p_e^s, p_c^s and p_b^s. Each plot represents one of the runs of simulation which, for networks of these sizes and variances, literally distributes the critical probabilities all over the place. But the plots of the largest cluster sizes always look very symmetrical. This seems to suggest that these sizes may represent the point of percolation better than the percolation probability.

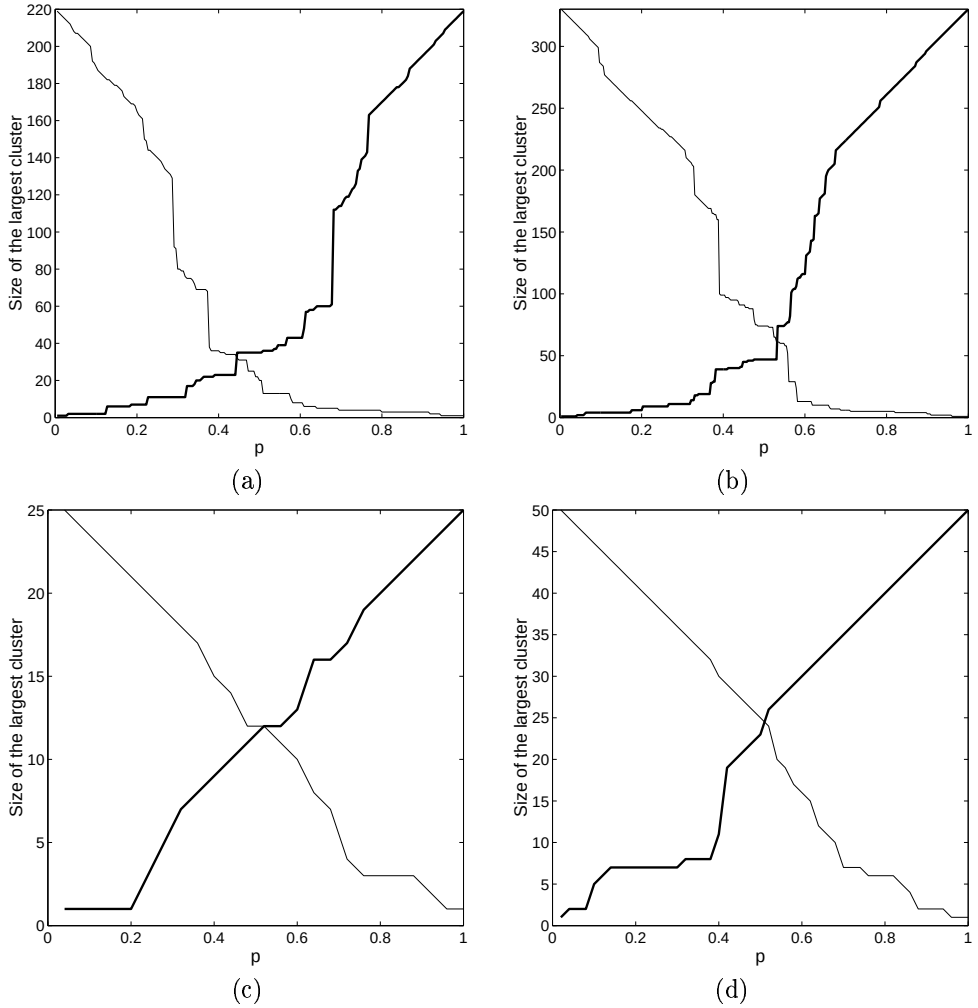


Figure 7.6 *Manchester. Plots of the largest clusters of,* (a) *vertices,* (b) *edges,* (c) *cells and* (d). *Here $(p_v, p_v^s) = (0.5455, 0.7182)$, $(p_e, p_e^s) = (0.6152, 0.4939)$, $(p_c, p_c^s) = (0.6400, 0.6000)$ and $(p_b, p_b^s) = (0.1200, 0.5000)$ respectively for the cases of* (a), (b), (c) *and* (d).

The mean percolation probabilities from (2×10) similar simulations as the two shown in each of the four cases of Figure 7.6 are $\bar{p}_v = 0.6723 \pm 0.0762$, $\bar{p}_e = \text{Er}[0.6083, 0.0838]$, $\bar{p}_c = \text{Er}[0.6240, 0.0921]$ and $\bar{p}_b = \text{Er}[0.3720, 0.1149]$.

Next consider Amsterdam in The Netherlands. The map for our purpose is shown in Figure 7.7. Areas shown in Figure 7.7 (b) are usually bound by main roads or the rims of the picture. There is no definite relations between vertices and edges on one hand, and cells and bonds on the other.

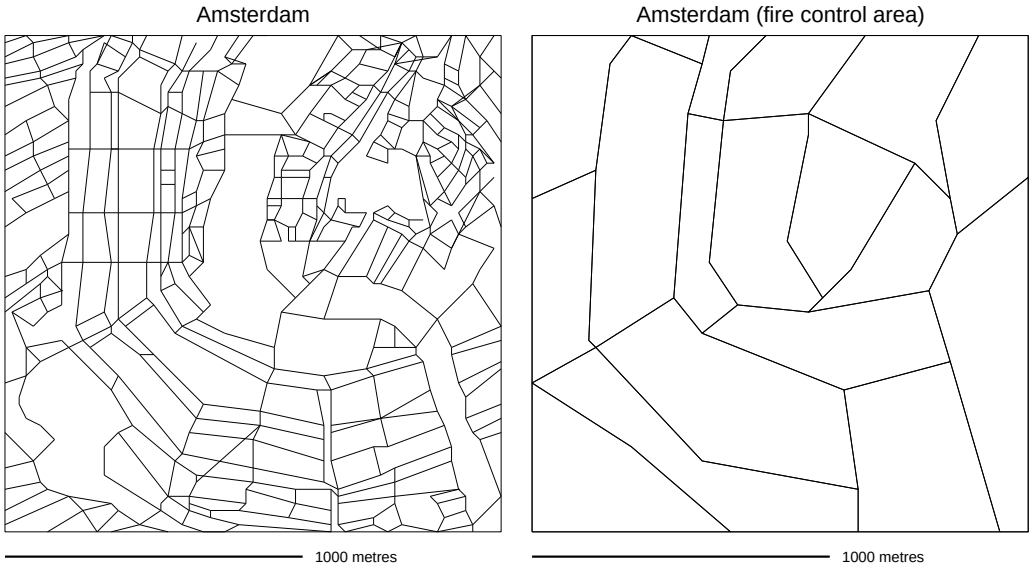


Figure 7.7 *Amsterdam in this picture has $n_v = 487$, $n_e = 745$, $n_c = 18$, $n_b = 35$, $x_v = 3.0513$, $x_e = 4.7007$, $x_c = 3.8889$ and $x_b = 6.8000$.*

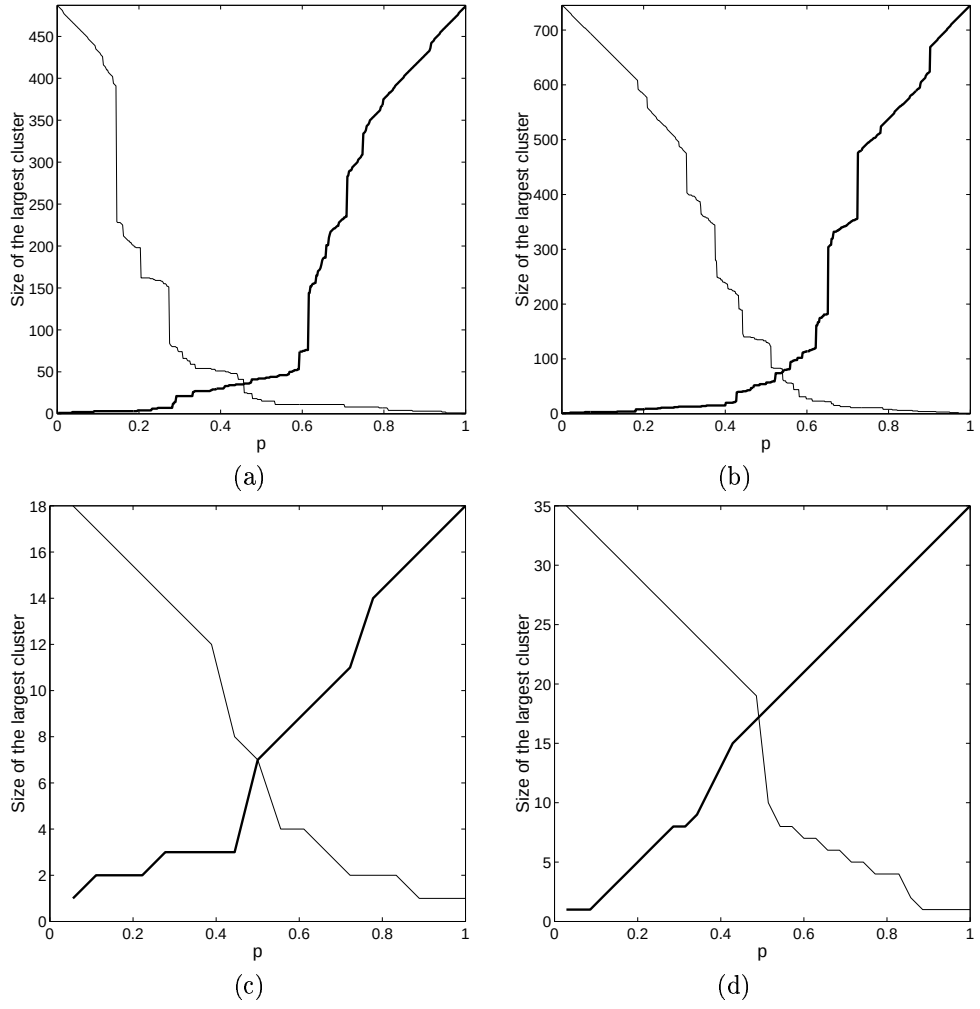


Figure 7.8 *Amsterdam. Plots of the largest clusters of,* (a) *vertices,* (b) *edges,* (c) *cells and* (d). *Here $(p_v, p_v^s) = (0.7187, 0.7146)$, $(p_e, p_e^s) = (0.6309, 0.5597)$, $(p_c, p_c^s) = (0.7778, 0.4444)$ and $(p_b, p_b^s) = (0.4286, 0.2571)$ respectively for the cases of* (a), (b), (c) *and* (d).

From (2×10) simulations we obtain $\bar{p}_v = 0.7374 \pm 0.0500$ and $\bar{p}_e = 0.6328 \pm 0.0595$, whereas

from (2×11) simulations $\bar{p}_c = 0.5960 \pm 0.1395$ and $\bar{p}_b$ 0.3922 ± 0.1395.

Then consider the road networks of Brussel in Belgium, as shown in Figure 7.9, to gether with the fire control area, and the largest cluster sizes in Figure 7.10. Fire control strategy is only one possible application to which the percolation of areas. Many other applications which are similar in nature, for instance emergency evacuation zones, earthquake evacuation zones, zones prepared as measure against a terrorist gas attack, *etc.* There are also other applications, for example strategic areas in market planning and the study of mineral deposits.

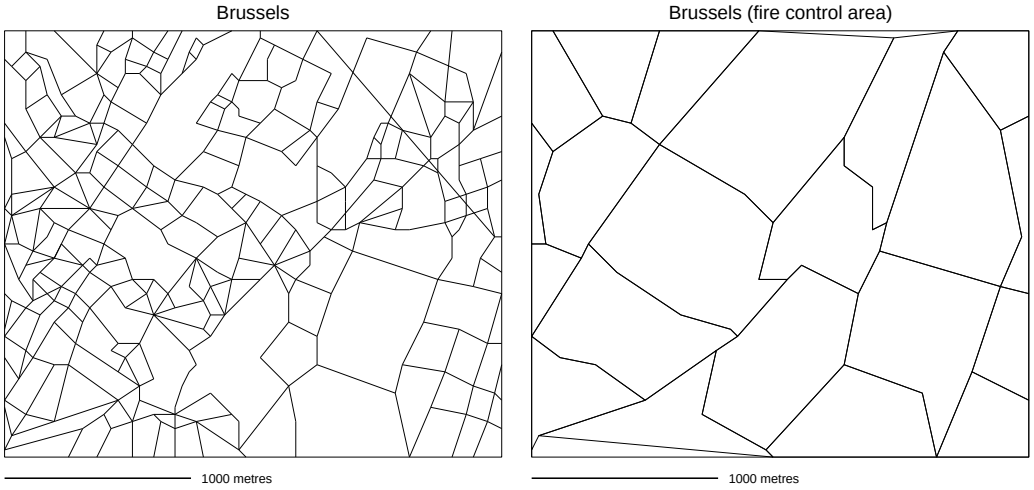
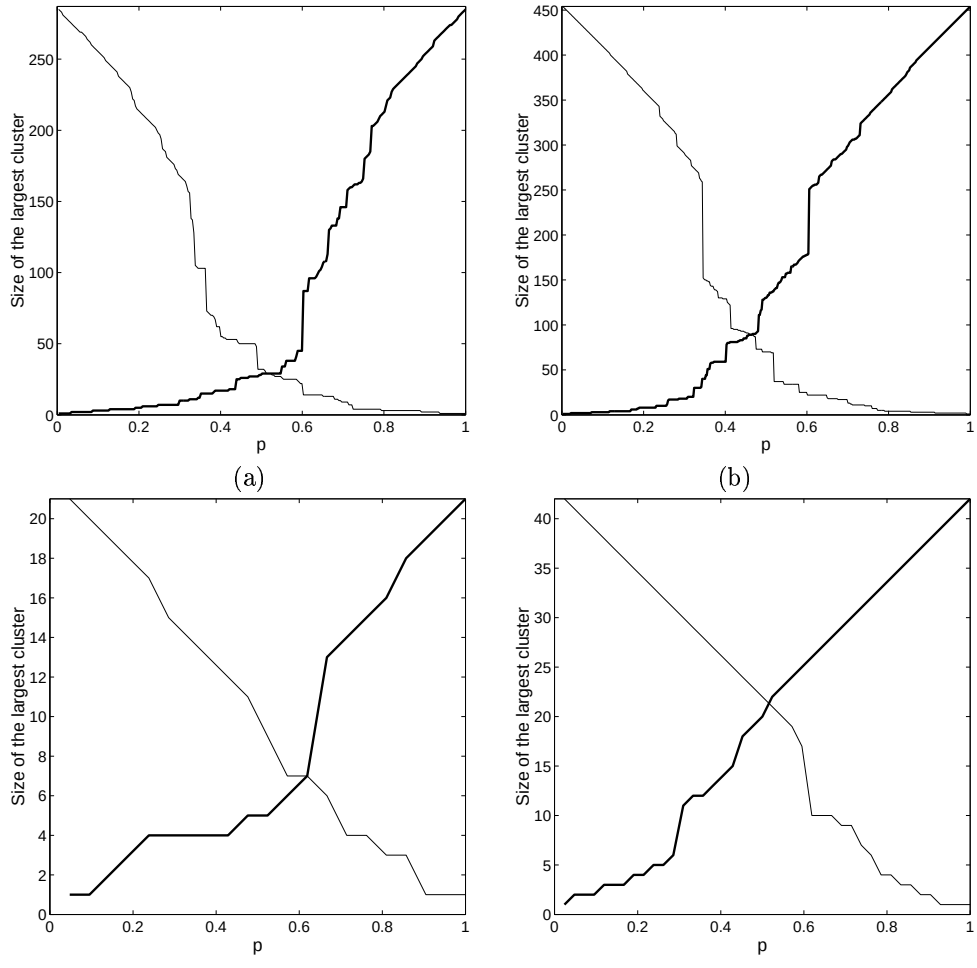


Figure 7.9 *Brussel has $n_v = 287$, $n_e = 454$, $n_c = 21$, $n_b = 42$, $x_v = 3.1568$, $x_e = 3.1568$, $x_c = 4.0000$ and $x_b = 7.0476$.*

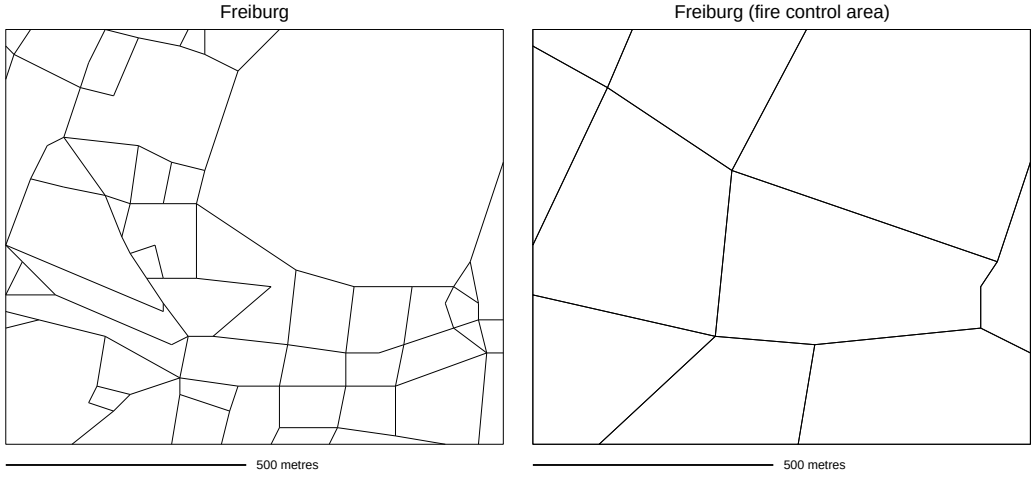


(c) (d)

Figure 7.10 *Brussel. The largest clusters when percolate by means of* (a) *vertices,* (b) *edges,* (c) *cells and* (d). *Here* $(p_v, p_v^s) = (0.6167, 0.6411)$, $(p_e, p_e^s) = (0.6057, 0.6586)$, $(p_c, p_c^s) = (0.6667, 0.5714)$ *and* $(p_b, p_b^s) = (0.3095, 0.4286)$ *respectively for the cases of* (a), (b), (c) *and* (d).

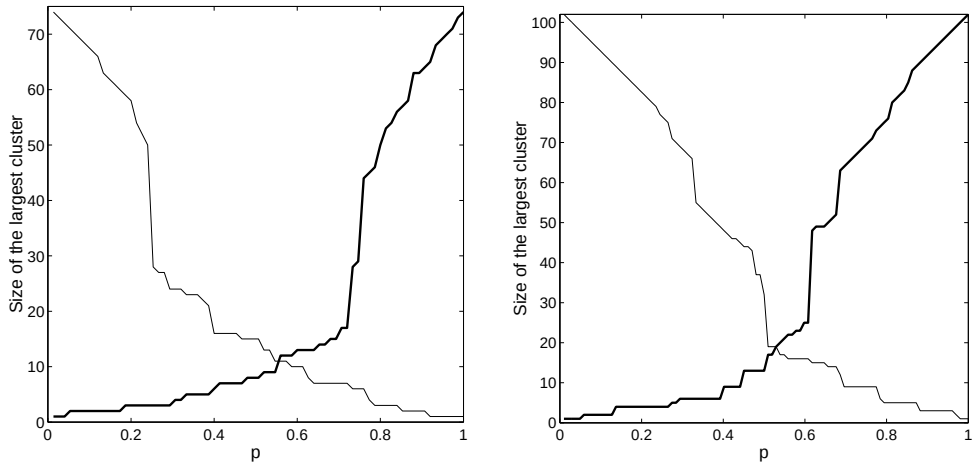
From (2×10) simulations on Brussel we obtain $\bar{p}_v = 0.6580 \pm 0.0771$, $\bar{p}_e = 0.6205 \pm 0.0540$, $\bar{p}_c = 0.6286 \pm 0.1043$ and $\bar{p}_b = 0.3786 \pm 0.0924$.

And then consider a small town Freiburg in Germany, where the many roads that are reserved for pedestrians only seem at a first glance to have altered much of the structure. But simulations have shown that the percolation probabilities remain comparable with networks of other towns. The area- and bond coordination numbers obtained for Freiburg are rather low compared with other towns. This could mean that the emergency properties of the town is different from those of others. Its lower connectivity could mean that it is more robust than others against an attack or in the face of catastrophy. But it could also make it more difficult to evaculate from an area. More precise relationship between the valence and the interpretation in terms of physical networks can only be possible by more extensive investigations in the future.

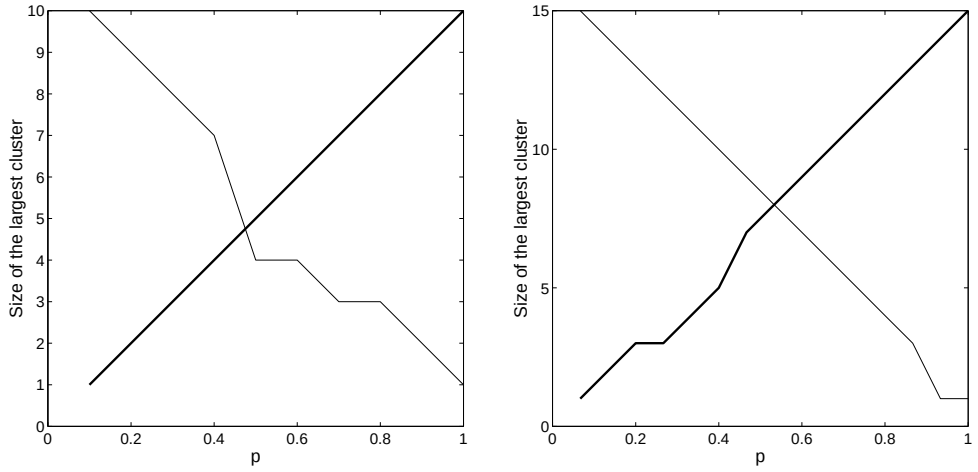


(a) (b)

Figure 7.11 *Freiburg has* $n_v = 75$, $n_e = 102$, $n_c = 10$, $n_b = 15$, $x_v = 2.7200$, $x_e = 4.4510$, $x_c = 3.0000$ *and* $x_b = 4.5333$.



(a) (b)



(c) (d)

Figure 7.12 *Freiburg. The largest clusters when percolate by means of* (a) *vertices,* (b) *edges,* (c) *cells and* (d). *Here* $(p_v, p_v^s) = (0.7333,\ 0.7733)$, $(p_e, p_e^s) = (0.6863,\ 0.6275)$, $(p_c, p_c^s) = (0.4000,\ 0.5000)$ *and* $(p_b, p_b^s) = (0.4000,\ 0.3333)$ *respectively for the cases of* (a), (b), (c) *and* (d).

Again, from (2×10) simulations we obtain for our Freiburg $\bar{p}_v = 0.7400 \pm 0.1214$, $\bar{p}_e = 0.6490 \pm 0.0787$, $\bar{p}_c = 0.5450 \pm 0.1276$ and $\bar{p}_b$ 0.3767 ± 0.1190.

§ 7.3 Theory of graphs and its applications

Gay and Preece (1975) study graphs on square lattices, represent them as matrices and then apply this to the study of networks of fluid distribution through pipes where there are branch flows and nodal pressures. Such a graph has n nodes, $m = b - n + 1$ meshes, where b is the number of branches, and any one of its trees contains $(n-1)$ branches. The direction of a mesh is the direction of the link connected to it. Define the branch-mesh incidence matrix $C = [C_T; C_L] = [C_T; U]$ and the node-datum incidence matrix $B = [B_T; B_L] = [B_T; 0]$, where c_{ij} is 0, 1, or -1 when the branch i respectively is not in the mesh j, has the same-, or opposite direction; similarly, b_{ij} is 0, 1, or -1 when the branch i is not included in the node to datum path j, is going away from, or towards the datum node. Then $V = E + e$, $J = I + i$, $V = ZJ$, $J = YV$, where E is the vector of branch pressure sources, *i.e.* pumps, e the vector of branch pressure rise, I the vector of branch flows due to the external inputs and outputs, i the vector of branch flows due to other causes. Then the mesh method has the solution $i' = (C_T^T Z_T C_T + Z_L)^{-1}(E_L' - C_T^T Z_T B_T I')$ and $E_L' = C_T^T E_T + E_L$, where $\gamma = [B, C]$, $V' = \gamma^T V$, $J = \gamma J'$, $J' = [I'; i']$ and $V' = [V_T'; V_L']$. The node method, on the other hand, gives $e' = (A^T Y A)^{-1}(I' - A^T Y E)$.

Operation research uses graphs to help in its search for optimum solutions. For example the Evolutionary Operation (EVOP) and its variant Rotating Square Evolutionary Operation (REVOP) use covering graphs on a body-centred lattice in two- or three dimensions (Lowe, 1964) – corresponding to the cases of considering two- or three factors respectively – in finding its solution. Simplex EVOP is another variation, where 2- and 3-factor regular simplexes are used, which has the advantage that it uses the least amount of experimental points. An n-factor regular simplex has $(n+1)$ points, in particular a 2-factor simplex is the equilateral triangle and a 3-factor simplex is the regular tetrahedron. All of these methods are possible at number of factors higher than three, but the procedure becomes more complicated and can not be shown graphically.

A graph is connected if it cannot be divided into two subgraphs without common vertices and edges. A linear combination of oriented edges, $\sum a_i e_i$ where $e_i = \pm 1$, is a simple cycle if a_i is 1, -1 or 0. It is a cycle if a_i are integers satisfying some linear relations, and a chain if they are arbitrary integers.

If vectors from the origin to vertices A_i follow the relation $A = \sum m_i A_i$, then m_i are the barycentric coordinates of A. Here $m_i \geq 0$, $\sum m_i = 1$ and A is the centre of gravity of the point masses A_i. A convex linear hull of A_i is then the space covered by changing m_i to give all possible A's. It is also known as a rectilinear-, Euclidean- or n-dimensional simplex. Simplices can simply be represented by the of their vertices. An n-dimensional simplex s_i^n has $n+1$ $(n-1)$-dimensional faces the *i:th* one of which is where $m_i = 0$. A k-dimensional face of s^n is the set of those points of s^n for

which $n-k$ barycentric coordinates are zero, while the rest change such that the corresponding m_i are non-negative and their sum is equal to unity. The number of k-d faces in s^n is ${}^{k+1}\mathrm{C}_{n+1}$.

The boundary of s^n is the union of all its $(n-1)$-d faces. The centre of a simplex is a point where $m_i = m$.

8

§ 8. Percolation theory as economic models

Even if there existed only a single economic catastrophe in the whole history of man one would have needed to come to a conclusion that it is governed by a percolation theory. In fact the number makes no difference, only the existence is of consequence. If only there had been but one that would have been one too many, but there are too many of these catastrophes for us to ignore their presence.

Economic and world histories tell us that the freedom possessed by each part in all scales of the social structure is a sufficient condition for the robustness in the economics and society. I feel that it is also a necessary condition.

These phenomena are like percolation. All existing economic theories are valid up to and not including the point where they occur. Therefore, if we must understand the working of our economic system we have to know it in the light of the percolational theorems.

Lack of freedom has already proved to be disastrous following the collapse of the Soviet Union and the eastern block countries. Deprived from their freedom, people and states alike will soon become poor in their facility to think because it had neither been practised nor honed. Interventionism has shown to be equally disastrous, as though we could measure the depth when they occur and compare, by what happened to Germany following the end of the First World War. The Bismarckian Empire collapsed in 1918 as a consequence of military defeat. The first immediately striking economic consequence of this was the growth of the public sector. Another change was that more of the government expenditure came under control of the central state. Then in summer 1922 hyperinflation set in as a consequence of the percolation which ultimately resulted in the Second World War.

The Roman Empire died because it depended too much on the central government in Rome. All the roads, no matter they led to Rome or away from it, only helped to increase the coordination number of the capital city of the empire. Every city in a robust empire, according to the percolation theory, has to be linked to its local nuclei city. With a single nucleus the network has only one cell, and that is too small for a network to be able to withstand any attack or to be robust. This is so disregarding whether the network is large or small, and in fact the larger it is the worse it will become. In this respect the United States, with its various states and local governments, is the best model that we have. The Brtish Empire, together with all other empires that are its contemporary, has already expired. And the ends of all empires in that fashion seems to say something similar to our idea of a robust structure. All of them had colonial territories, but non of these were treated as equals by the ruling countries. The European Union could fare better if it learned from them, as well as from mistakes that the U.S. had made.

§ 8.1 Object-location

The object location problem is geometrical in nature. My design exercise (Tiyapan, 1995, KNT2(ii)) during my MSc course, supervised by Dr. Zarrop of the Control Systems Centre here at UMIST, is included here in § E.4. Figure 15 to 17 which replace the original ones are produced by the original respective programs with minor correction, namely in the drawing of the lower two sub-figures in each one of them. In addition, Figure ℵ and ℵ give an idea where the mask is relative to the object at each instance of time.

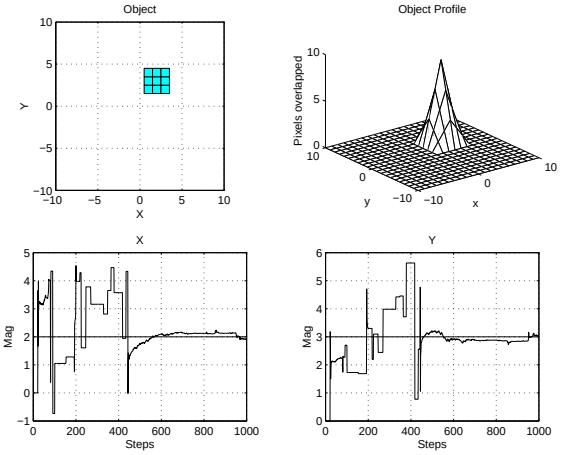


Figure 8.1 is the object location of a square object by a square mask of equal size. Figure 8.1 uses the same program which produces Figure 15 of § E.4. It shows the square object, the hill derived from the area of intersection with respect to the position of the mask, and the x- and y-axes of the mask with respect to time.

Figure 8.1 Square object located by a square mask..

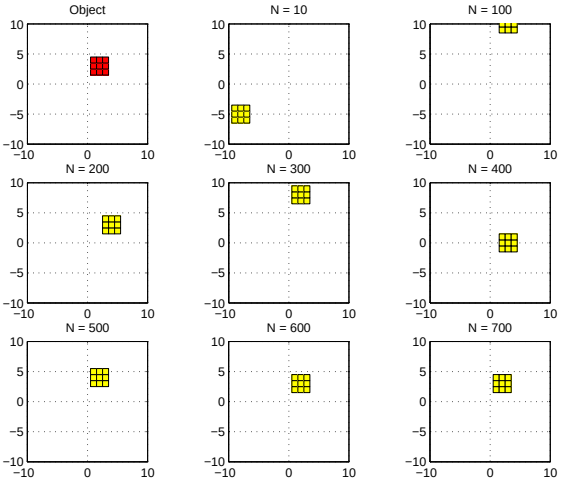


Figure 8.2 shows the object together with eight snapshots of the mask at different time. While doing a random searching the program may at times think that it has found the object somewhere and jumps to that location, only to realise later the mistake. This is due to the interference by the noise. On the other hand if it knows that it is doing the random searching, it retains the latest position that it thinks is the correct one. All of this is shown in Figure 8.1 and 8.2.

Figure 8.2 A snapshot of of the object and eight others of the mask..

The paper reproduced in § E.9 has been submitted to be considered for presentation at the 7$^{\text{th}}$ International Symposium on Dynamic Games and Applications which was to be held in Japan on 16–18 December 1996. In April 1996 I submitted an extended abstract of the same to the organising committee and it was accepted. I thereby emailed to Dr. Zarrop to ask whether he would mind being a coauthor of the finished paper since it was him who gave me advises on the work which was the product of the design exercise course that I did with him, he said he would be happy to be one and I wrote to thank him. But upon seeing the finished work he wrote to tell me to be its sole author instead because he sees nothing new in it and, without giving suggestions on how it could be improved, rather have nothing to do with it. I duly sent the manuscript with his name taken out in July and it was not accepted.

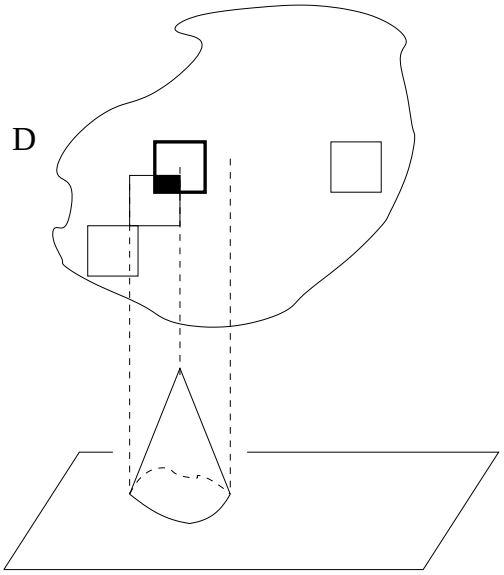
A work by Tiyapan (1996, KNT2(ii)) on object-location problem is reproduced in § E.14 because the original manuscript and results is irretrievably lost. It appears in the proceedings of the ATACS-96 conference. This conference is one among the series of mock conferences under the name of *Advance Theory and Application of Control Systems*, which is organised each year at various places in Japan by laboratories which are historically of the same family. ATACS-96 was participated by 14 research groups or laboratories from six different universities, with altogether 102 participants, 48 of whom gave a presentation. ATACS-97 has 42 paper presentations and 130 participants from 16 research groups of altogether ten universities. ATACS-98 was participated by 114 people from

13 research groups of five universities, among whom 38 people were speakers. These are the three ATACS conferences I attended, namely the one in 1996 at Izu Peninsula in Shizuoka Prefecture and the one in 1998 at Lake Kawakuchi in Yamanashi Prefecture both of which are in the excellent setting of the Fuji-Hakone-Izu National Park, and in 1997 at a youth centre bordering on the lovable Yoyogi Park in the bustling Shibuya District of Tokyo.

To summarise, the RLS technique with random searching employed in § E.14 is developed further to accommodate back-stepping which helps produce a better result as shown in § E.17. The latter paper has been presented at an international conference in Singapore (Tiyapan, 1997, KNT2(ii)) and then as a seminar at TIT on 3^{rd} January 1998 . The former is the second joint conference of the International Simulation Societies called World Congress on Systems Simulation. It was organised by IEEE Singapore Section, the Society for Computer Simulation Europe, and the Society for Computer Simulation International. I gave a presentation on the last session, Session 21 on Potpourri, of the last day of the conference. As I did not buy the conference proceedings because it did not contain my paper except as inserted addendum pages copies of which I already had, I have no idea what the papers of the other presenters look like. The paper has been reviewed by two reviewers both of whom gave it a good for technical quality and an average for importance. For both the originality and the relevance to WCSS97, one gives a good while the other an average. The paper has been unanimously marginally accepted with the readability and presentation scoring somewhere between below average and average. One reviewer commented, 'The abstract is poor and too short. The set of references is also very poor. There is no conclusion to the paper! Previous remarks affect the quality of the paper, but it still remains in [sic] interesting study. The algorithm of the last section should be structed [sic]. Considering what I see of the paper I suggest that the author use a classical chart to present his algorithm. More precision (description) must be added to the functions an [sic] variables in equations. Some are described some are not. Before the algorithm section the author should give more precision concerning the random search and he could add references. Though my english is not very good, I think that the text should be improved (have it re-read by a native or eglish teacher if possible, or at least by another colleague.[sic]' The other gave a more succinct comment, 'The paper seems to be in-complete. The problem under consideration should be defined more clearly. Advantages of the proposed algorithm should be discussed in more details. Practical problems that could be solved by the proposed algorithm should be discussed. Conclusions should be added.' None of these suggestions I had followed because Dr. Helen D. Karatza the Chair for paper review wrote saying that IEEE Singapore would inform me about the final decision for my paper but I heard nothing from the latter although they told me later that they had sent a letter to me at the Furuta Laboratory, TIT. Interestingly enough I find a note in Japanese received from Ms. Shinata dated 7^{th} March just now among the document in I keep. She was then the secretary of Professor Furuta. The note says that he could not help with the cost of the conference since he has no idea what kind of paper I had written. Only now that it occurs to me that a few month earlier on I did actually present a similar paper which contains two-third of the contents in a conference which was organised by himself (*cf* Tiyapan, 1996, KNT2(ii), or § E.14)! That aside, I have thus far come to the conclusion that the purpose of the organiser of a conference is in order to make money.

The following summarises my presentation at the WCSS97 conference. The original ideas of the research *RLS algorithm for object-location problems* come from an example in a design-exercise for an M.Sc. course in Control and Information Technology which I did with Dr. M. B. Zarrop at the Control System Centre, UMIST, during 1994–1995.

The same paper I also presented in a seminar of the Furuta Laboratory on 3^{rd} January 1998 . The manner of the presentation is recaptured in the following. The performance function of an object location problem can be represented by the area of the intersection between the object and and a randomly placed mask as shown in Figure 8.3 and 8.4. The former shows the hill positioned relative to the centre of gravity of the object, while the latter shows the contour of the hill resulted from a square object.



Both the object and the mask are within the domain D. The mask is placed in such a way as to maximise the area of its intersection with the object. The object and the mask may have any shape, and their shapes can differ.

Figure 8.3 The hill in an object location problem..

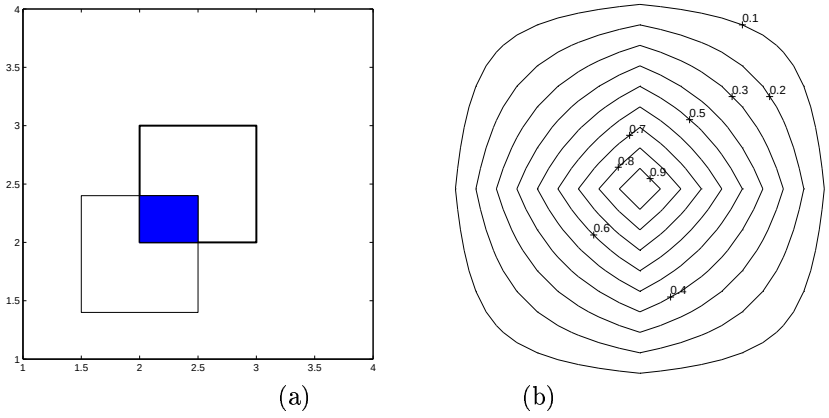
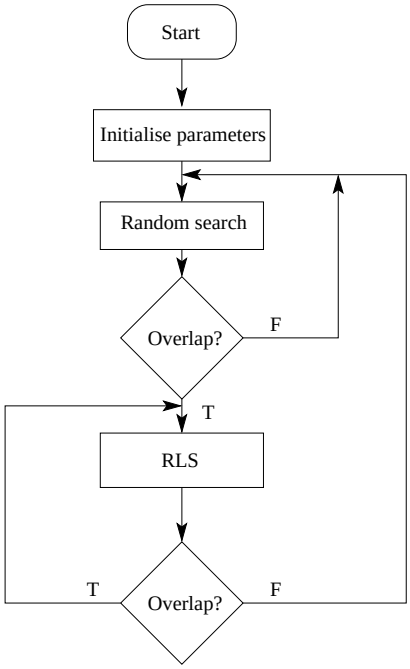


Figure 8.4 (a) *Area of the intersection between square object and a square mask,* (b) *contour of the hill thus obtained.*

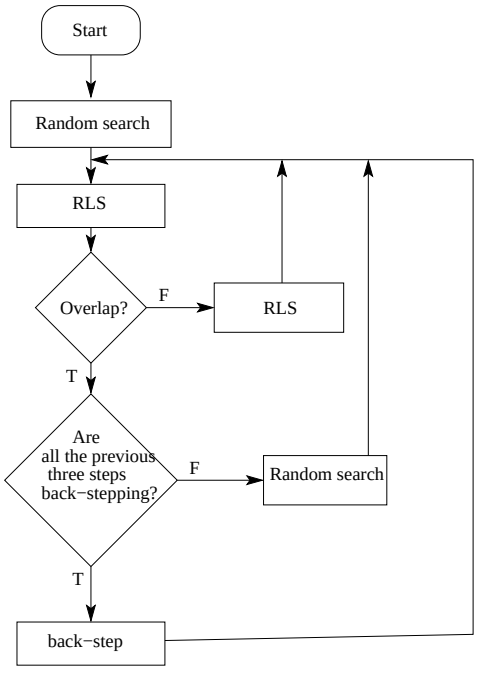
The procedure used in the paper I presented at the ATACS96 conference is shown in Figure 8.5.



This procedure does a random searching whenever the position of a hill is lost. Once the hill is found it switches over to a recursive least square subroutine.

Figure 8.5 RLS procedure with random search.

Then Figure 8.6 shows the improved algorithm in the paper presented today.



This is the same as the procedure in Figure 8.5 except that whenever losing sight of the hill it recursively steps back a step at a time for the maximum of three times or until the hill is found again. If the hill is still not found then it switches to do random searching as the other procedure does.

Figure 8.6 RLS procedure with random search and back-stepping..

The simulation mentioned earlier on, namely where Figure 8.1 is, has the methodology as shown in Algorithm 8.1. Here σ_δ^2 is the dither signal variance, σ_ν^2 the noise variance, ρ a random number, O the object, I the image, I(n) a square identity matrix of dimension n and d the dimension of our space which lies between $\pm d$ in both directions. The program in § A.1 is based on this algorithm.

Algorithm 8.1 *Object location algorithm.*

Create the object;
for each step **do**
  $\delta \leftarrow (3\sigma_\delta^2)^2(-1,1)$;
  $p \leftarrow -b_1/(2a_1) + \delta$;
  $q \leftarrow -b_2/(2a_2) + \delta$;
  $\gamma \leftarrow O \cap I$; $\nu \leftarrow \sigma_\nu(0,1)$;
    $y \leftarrow \gamma + \nu$;
  if $y > 1$ **then**
    $x \leftarrow [p^2, q^2, p, q, 1, 1]'$;
    $t \leftarrow t + 1$;
    $p \leftarrow -b_1/(2a_1)$;
    $q \leftarrow -b_2/(2a_2)$;
    $P \leftarrow p$;
    $Q \leftarrow q$;
    $\Gamma \leftarrow \Gamma\left[\mathrm{I}(6) - (xx'P)/(\lambda + x'Px)\right]/\lambda$;
    $\theta \leftarrow +\Gamma x(y - x'\theta)$;
    $a_1 \leftarrow \theta_1$;
    $a_2 \leftarrow \theta_2$;
    $b_1 \leftarrow \theta_3$;
    $b_2 \leftarrow \theta_4$;
  else
    $t \leftarrow t + 1$;
    $p \leftarrow P$;
    $q \leftarrow Q$;
    $b_1 \leftarrow -2a_1(-d, d)$;
    $b_2 \leftarrow -2a_2(-d, d)$;
  endif
endfor □

§ 8.2 Economic modelling

My interest in the modelling of economic systems started immediately following the devaluation of the thai Baht in July 1997, which in turn triggered the economic collapse of east Asian countries. But the seed of this interest may have been germinated nearly seven years earlier when I went to Budapest for a job training through a program by the *Association Internationale des Etudiants en Sciences Economiques et Commerciales*, aka AIESEC. It was there that I experienced firsthand an economic system with inherent potential, but which was ailing and struggling to survive. It was also there that I have come to know the word *hard currency* and learnt about terrible consequences to people in a country whose money is otherwise from being hard. Some of these consequences are the proliferation of black markets where money is illegally traded for hard currencies like the US Dollar or Deutsch Mark, and the ever soaring inflation rate. In Poland at that time, for example, the inflation is above 700 percents. In such an unstable society stealing is a matter of fact in everyday life. Robbery becomes so widespread that the police no longer bother to catch robbers, and this is so much a more decent thing to do since otherwise there would have been no place to put them in in the first place. This I know because it was in Poland around June 1990 that I was robbed of nearly all of my personal belongings. Afterwards when I have become initiated to the theory of percolation, I ascribe the transition between a hard and a soft currency to the percolative process. Now I still see this, as well as the changeover of the eastern bloc countries, as a phase transition.

My brief presentation at a conference in Japan (Tiyapan, 1997, KNT2(ii)) has been warmly received by some of the experts who attended, despite criticism from a member of the organising committee before the presentation. My belief was, and still is, quite different from the conventional theory as to the cause of this national mishap. Instead of seeing the attack on thai Baht made by an american hedge fund which is run by George Soros, I put most of blame on other seemingly unrelated factors that deprive Thailand of essential robustness in many of its fundamental structures, particularly on too much centralisation towards Bangkok and the percolated traffic networks in that city which came close to a standstill. Others argue that the Soros hedge fund represents a corporate crime against a nation because no matter whether it is the true cause of the catastrophy or not the intention of attack was there. But such is not the kind of matter which concerns us here. In essence, I give all the value to the structural soundness of a system and the infrastructures and none to any single occurence or mistake in management decision. A country, I argued, will run sound if if it has a potential to do so, and not otherwise. The potential of a country depends on such things in the background as all the infrastructures including transportation and telecommunication systems, education, forest, and decentralisation.

The self-sufficiency practice recommended by King Bhumibolof Thailand in his annual speech to the thai people on his birthday in the December following the crisis is the very core idea of decentralisation in disguise, namely that of local management which will ultimately lead towards local government. In his speech he told the Thai people to live a self-sufficient life, to grow vegetables in the backyard enough for one's own need, and to sell and make a business out of it if one has more. I believe his idea goes deeper and his teaching more a parody from the necessity from him to remain impartial even in the face of a crisis. In short, the self-sufficient of oneself will lead to that of one's family, village, town, district, province, and finally the country. The Constitutional Laws have been changed to favour this, which I think is a crucial step.

There has been a complete turnaround in the policy in Thailand now compared with what had been before the 1997 crisis. Then it was everything to be centralised in Bangkok. Every prosperity, you name it, must go to the capital, and every policy out from it. It had been so for a long time, for many generations, and democracy did not much improved it. When Smyth (1898) lived in Thailand in the 1890's he saw what I had seen one century later in the 1990's, that everything beautiful in Bangkok comes from, and at the expense of, the rest of the country which is left to be poor. Anyone travelling in the city even today can see quite easily that Bangkok by no mean represents Thailand. With this kind of policy, no doubt everyone tries to get to Bangkok. This worsen its problems further. Nothing that I may say on this point can be better than the following account made by Smyth from the work already quoted.

> I am certain that the Government in Bangkok have never realised the disastrous effect which the system had on the distant province. The revenue each year was maintained at the proper figure, and therefore no questions were asked. The money went to Bangkok, and the P. W. D. spent it in improving the look of the town. Visitors were impressed with what was being done in Bangkok, and went away persuaded that Siam was an enlightened country. The poverty-sticken condition of the provinces did not come under their notice. 'That's all our money,' cried a disconsolate provincial Government to me, spreading wide his fingers towards a canal in Bangkok beeautifully embanked, with a charming road on either side. 'I have had to roof my court-house at my own expense.'

Fifty years on, the canals in question started to deteriorate and fifty years later either have been filled to make more roads or left a dirty sewer. In 1992 there was a plan made by the Bangkok Metropolitan Administration to construct wastewater treatment plants around Bangkok. The construction was to be in two phases each of which worths 5,000 bahts, and equivalent of 120 million pounds sterling. At that time I was working at the Loxley International Public Company in Bangkok in the capacity of a sales engineer. In the second bidding for this turn-key project I was transfered to the Business Development Division to work as the estimator of the project, reporting directly to Mr. Ronald James Savage who was one among the Board of Directors of the company. Loxley joined force with several consultants and contractors to bid for this project, and as the company was to provide all the necessary equipments and technologies my job was to search all corners of the world for everything everyone would need. It was chalenging, but then I was a workaholics so it was all I needed. In the end we did not win the project, however, so that is as much as I have to say about it except for one thing, that is that I think the wastewater treatment project came far too late and was still not completed when the 1997 crisis occured. Seemingly unrelated, in the present argument and together with other things such as traffic, education, decentralisation, it has got everything to do with that crisis.

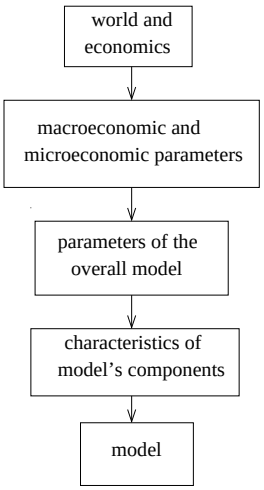
Now the policy has turned into that which lets the remote parts of the country manage their own funds and make their own decisions. There is, for example, a scheme of *a product a village* that fosters local artisans. As a result, local heritage becomes fad where is only five years ago it was fiction, something to forget, not to think about. If there is no hiccough in this process or a *coup d'état* of some sort, then the policy of Thailand now is by far sounder than it ever had been. Therefore Thailand to me now is in a better shape than it used to be when it was still dubbed a *tiger or the East.* However people often think otherwise. I always find myself involved in a controversial or difficult issue, and on the wrong side thereof. Before the crisis, for instance, when most people looked up to America with hopeful eyes I always said to my colleagues, 'we should not put our whole faith in America', that Thailand should not rely too much on any other countries, and I was looked at questioningly. After 1997 I said, 'it's not half the fault of the Americans or anyone else but us', and I was eyed with disgust. The hedgefund which attacked the Thai Baht in 1997 was only an instrument, not the reason, for what happened thereafter. It answer the *when* question, but the more important *why* question has already been answered long before that (*cf* Smyth, 1898). I can say this because as easly as 1990 while the Thai economy was in the bull I wondered myself why it has not collapsed because, as I often said when discussing with my colleagues engineering students, according to my reasoning it should. Immediately after the disastrous event while the country was in the blue and I was in Japan I told myself that I had already known why it happened, now I knew *when* and *how* it did.

What followed was a period of a few years when the whole country, noticeably Bangkok, was so quiet you would not believe. For some times the country was a paradise for travellers. Everything was cheap and nothing was not on sale. Instead of increasing, I felt that on the whole crimes had reduced, perhaps because such tragic events always shook the hearts of people, the thieves' and the

gentles' alike. The Tourism Authority of Thailand coined this marvellous word *Amazing Thailand*, to promote tourism in Thailand abroad, which fitted the mood of the country like a glove because everyone felt like staring wide-eyed with disbelief at what had happened and what was happening around them. I was equally amazed at what was happening around me when I visitted Thailand from Japan, but not at what had happened because, like the next great Tokyo earthquake, I had been expecting it for quite a while before I got tired and stopped thinking about it.

An analogy of distributed government can be seen in computer science under such terms as distributed systems, parallel processing, and object-oriented programming. I see it as essential, if not as the only way, towards a sound and robust economic systems.

At this ATACS conference in Tokyo in October 1997 I explained economic modelling as follows.



The economics of the world can be represented by macroeconomic and microeconomic parameters. Together these two make up the parameters of the overall model and describe the characteristics of components of the model.

Figure 8.7 Economical models..

Later that year I submitted two papers (Tiyapan, 1997, KNT2(ii) and KNT2(ii)) electronically to a journal, the American Journal of Physics I think it was. For the reason that both papers were products of my personal research interest which has nothing to do with my commitment to the Tokyo Institute of Technology, I could only spend a meagre amount of time on them. As a result, the work was too preliminary and there was no reply from the publisher except the first computer-generated message saying that they have received my files. No matter how poor my papers were, I felt that to get a negative feedback from an editor was already bad enough, but to get no reply was worse.

In the light of the argument above, what Hammond (1998) called a *Market World* corresponds to the case where the globalisation expands smoothly and permeates all countries from developed to third world to form a single unbroken cluster. What he called a *Fortress World* corresponds to the world where the the free market only benefits some countries while the rest is left deprived. The poorer countries encroach on the richer ones like water encroaching the residual oil in an oil reservoir; islands of prosperity surrounded by seas of poverty. Yet his third and last vision or version of the world, the *Transformed World*, would correspond to a changed structured of the network of culture and economy which gives a more robust system and accommodates all the needs of the future generations. Among the things that are changed in this case are the cultural norms and values.

Cultures are like languages. Like switching from one language to another, when travelling to another country one switches from one frame of mind into another. Unless this is properly done there will always be some adverse symptoms to experience, and *culture shocks*, which are common in novices, become less so in experts. In the languages of cultures, when we deal with the Japanese's we should keep as much a polite manner as possible and take our responsiblilities to the other party as seriously as possible regardless to what the small prints might or might not say. On the other hand if we are to conduct our business with the Americans, for instance, we should question every move. Generally speaking while the West favours contracts, the East has been run mainly by personal relations. The catch came when the latter party adopted so many of the ways and ideas of the former, and these got improperly mixed together with its own ways of thinking which are different to produce an incoherent, unstable and insecure structure. Stevenson (1895) when travelling *out of himself* to the United States in 1879 came across a mexican who was scruple enough to knowingly sign an imperfect paper for fear that the other party might think that he doubt his words otherwise.

The scruple sounds odd, he said, because we "have been brought up to understand all business as a competition in fraud, and honesty itself to be a virtue which regards the carrying out, but not the creation, of agreements." This no doubt answers the question why the 1994 economic crisis in Mexico occurred.

Because there is more to the economic well being of a country than economic figures can tell, one needs to look around in order to to find clues to what is going wrong. For this, the richest source is in the literary literatures, some of which had seeped out through the unconscious minds of some of the most experienced and sensitive minds. Also, at least in the field of economics the study of science and the social study should go along side-by-side. More than economic figures and those things which meet the eyes, studies in this area should take into account hidden factors, for example education, centralisation and policy in general, and turn them into a form which can be numerically or theoretically studied. Centralisation always adversely effects the economics, for instance the greater control by the central state of government expenditure as well as growth of the public sector Germany immediately after the First World War is followed by the hyperinflation which began in summer 1922 (*cf* James, 1986).

Descriptions of economic crises usually resemble percolation, eventhough or especially when they come from someone who has no ideas of percolation in his mind. For example, Rodelet and Sachs (2000) think that small events at times have large consequences. They talk about crisis of success, that both the collapse of Mexico in 1995 and of Korea in 1997 came after a prolonged period of euphoria. In Thailand, for instance, indicators in the third quarter of 1997 suggested nothing like a financial meltdown. On the other hand, while Mlesi-Ferretti and Razin (2000) mention a 'perfectly forseen speculative attack' as the thing which triggered the crisis in this case, there is no question about the network having already greatly weakened before the attack.

One has to distinguish things which trigger from things which cause. One can be very much certain that crises of this scale must come from a very slow process which works on the structure within. It may be true that delaying bubbles from bursting this way even results in a deeper crisis afterwards.

9

§ 9. Addenda

§ 9.1 Sociology

The nature of discoveries and progresses in science is according to Bacon (Francis Bacon, 1620) *a birth of Time rather than a birth of Wit.* This is the same idea of percolation and the description he gave is the very picture of the theory. According to him major scientific progresses come in revolutions which are sparsely distributed in both time and regions. There have only been three periods of major progress out of the *five and twenty centuries over which the memory and learning of men extends*, namely the Greeks, the Romans and the nations of Western Europe. These are narrow limits of time, the periods in between of which are unfavourable to development. A discovery or an invention, then, comes as a chance accident in the scale of an individual, and as a certainty when looking from a distance.

When the time is right and all the hidden momentum built up, theories will come on by itself as a rule. This does not negate the excellence of an individual, but in a society where there are enough multitude of individuals the show will always go on, with or without a particular genius. This idea can very well explain cases of multiple discoveries. According to Kekulé in his *Benzolfest* speech in 1890, when he ascribed his conception of the cyclic nature of Benzene in dreams, *certain ideas at certain times are in the air* and *if one man does not enunciate them, other will do so soon afterwards.*

To see the relationship of this with percolation it is possible to look at two different things in turn, first at the discoveries and then at the discoverers. With a unit being that of a *discovery* the connection to percolation is that big discoveries come as connections of other smaller and less obvious ones. A theory often has more than one perspective, and which one of them comes to the fore first depends much on which combination happens to percolate through first. The discoveries of Schrödinger and Heisenberg in Quantum Physics can bear witness to this both in the *combination* and the *multiple discoveries* parts of this argument.

Let us turn our attention now to the scientist and look at the one who does the discovering instead. The theory of percolation tells us that at the point of discovery he is by no mean the sole

integral ingredient. If he does not do it, then someone else will certainly do. In order to see this, I did four simulations for the cell, bond, vertice, and edge percolations on a two-dimensional Voronoi network and then another four with the same respective blockage of each case but considering the inverse phase instead. The number of units considered for the four cases are $n_c = 200$, $n_b = 416$, $n_v = 298$, and $n_e = 426$. With the order of simulations as described above, at just one step before percolation occurs there are respectively 10.6, 13.4, 11.5, 1.1, 11.24, 10.0, 19.4, and 7.9 percents among the remaining units which will readily trigger the onset of percolation. In other words, these are atoms which are able to link up existing clusters and form a percolating cluster.

The formation of mobs is an interesting phenomon comparable with phase change in physice. What happens is that an agglomerate of individuals becomes one and a single creature, the underlying mechanism of which still baffles any effort towards understanding it. Likely enough it has got something to do with psychology and the mind. But to me at least, the phenomenon is percolative. Having gained some acceptance from my previous writings (Tiyapan, 1995–1996, KNT2(ii), KNT2(ii) and [kbukt], respectively §'s E.5, E.6 and E.7) I gave my new work which briefly discusses the mechanism underlying the formation of mobs to an editor of the Sakkayaphab journal whom I know. At that time a political turmoil unequal in its degree and extent has been going on for five years. Whether by fate or by design there has been a successful but tragic use of mob in Bangkok. The word *mob* has joined the list of those synonym to *distrust*, namely *communist* or, in western community now for that matter, *Islam* and *evil*. Whether because of this or something else, the article (Tiyapan, 1996, KNT2(ii), or § E.15) simply and mysteriously got lost; no one would admit having seen it, and the translation of another subsequent article of mine (1996, KNT2(ii), or § E.16) has not been without a noticeable negligence. Thus to me distrust is also percolative. The list of things one finds over-distrusted without reasonable explanation goes on indefinitely, homosexuality, communism, *etc.* The same seems to be the case with bad habits. My father used to teach me using the following poem,

> Bad habit gathers by unseen degree
> like brook makes river, river runs to sea.

Looking back, it could have been the title of that article, *on pragmatists and idealists*, which has somehow convinced the editors into believing that it was political which to me is nonsensical. I only meant literary, even if at times philosophical. I include it here because it contains a curve showing a critical emotional transition.

The formation of the United States, the European Union, or the Commonwealth comes from the trust which acts to join countries together like glue boxes in TEX. Like all binding forces, trust is mutual and spreads in the same way as a growing cluster does. The cluster grows bigger as one or more members are added, and it becomes stronger as the level of the mutual trust increases. In a similar way, distrust is also mutual and also spreads. If I distrust you and you distrust me, I will make sure that I remain as far away from you as possible while you will certainly avoid me by all means in return.

Only these two are possible, so there are only two phases to consider, that of trust and distrust. The relationship where one trusts while the other distrusts would not be stable, since the former will soon learn to join the latter. Trust forms clusters of one phase, distrust another. The size of these clusters vary in a way similar to those in percolation of geometrical networks. The strength of the glue is analogous to the probability either of becoming or remaining a member of a cluster.

The rise of dictators, the proliferation of weapons of mass destruction, *etc*, these things I believe are the products of changes of something hidden within the underlying structure. Unless we find out what is happing in the background, these things will unavoidably occur. I believe that this unseen thing behind the scene is governed by some phenomena similar to that of percolation. I think that the key towards understanding many unexplainable phenomena is to investigate, in the light of the percolation theory, the working of agglomeration of countries or states like those of the United States and the European Union.

The believe that scientific discoveries are a birth of time, rather than of wit (*cf* Larsen, 1993), is the same as the idea of percolation. We know because we remember. And all the various discoveries of our time together with the knowledge we possess of the past bring us closer to another discovery. Scientific discoveries, then, is the collective product of humanity rather than property of a single person (*cf* Merton, 1965).

And because all species are also the product of percolation in time, our knowledge and consciousness, too, are the product of the universe. One may say that it is a personification when we refer to a collective noun, for instance a mob, as an individual. But the truth is that, under the

percolation theory, it is in fact a separate individual without any need for the use of a simile. The renormalisation group theory tells us that there exists a structure in a bigger scale that behaves like the individual components that comprise it. It seems, therefore, that for humans these collective beings of ours are still primitive compared with each of us as an individual. This is the reason why, whenever we come together, we alway make wars. In our case, then, we seem to be conscientiously percolated only individually not collectively. In the case of bees, on the other hand, it is the other way round. This is why a colony of bees does things which make far better sense than a bee does. But one can not say that even a *colony* of bees has consciousness, because there seems to be no morals in what it does. I do not know whether there are other beings in the universe both the collective and individual beings of whom have percolated conscientiously. But I believe they exist, in which case they should be more intelligent than us, though this is by no means necessarily the case.

§ 9.2 Control systems

The following are relevant subjects in control systems. I have one first degree and one master degree in the subject. From 1995 to 1999 I did a doctorate course in Tokyo but decided to quit after three years. Research in this area is still one of the topics that I would like to do in the future. One year before leaving Japan my research results must have look quite well because on 22^{nd} April 1998 my supervisor Professor Katsuhisa Furuta wrote me an email which says, 'Your reports are very nice! Excellent results. I am verymuch [sic] impressed. We can talk the results.', signed 'Furuta'. And again on 2^{nd} June 1998 another email, 'Dear mr.kit [sic] please come and explain your synchronous motor. Where shall we introduce the controller?', signed 'Katsuhisa Furuta'.

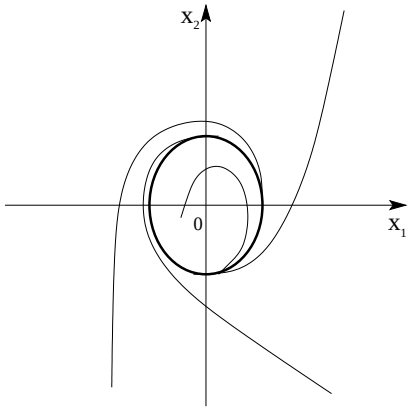
Although it is true that control systems is always used in military and misiles, one must not forget that it is also important for travelling in space and is likely to be of a great help to those space settlers searching for habitable planets in the future. Control system is used much in electrical, mechanical, and chemical engineering, with the typical time constant increasing in that order.

This section is in a way a brief recapture of what I did during my PhD study at TIT, Japan. I collected my works in a form of Technical Reports and gave a copy each to Professor Furuta who was my supervisor then. Sometimes when there was a spare copy left I would give it to another senior staff who worked in the Minami 5 building where the Furuta lab used to be. The computer files of these reports is no longer available even to myself. A great number of figures from simulation results in these reports are not reproducible here without the Simulink facility on Matlab. The first of such report was dated 16^{th} July 1998 but the works it contains started around the beginning of April of the same year. I studied systems listed in a book by Khalil (1996). One of these systems (Exercise 1.17 (4)) is

$$\dot{x}_1 = x_1 + x_2 - x_1\left(|x_1| + |x_2|\right), \qquad (\mathbf{34})_{\text{ix}}$$

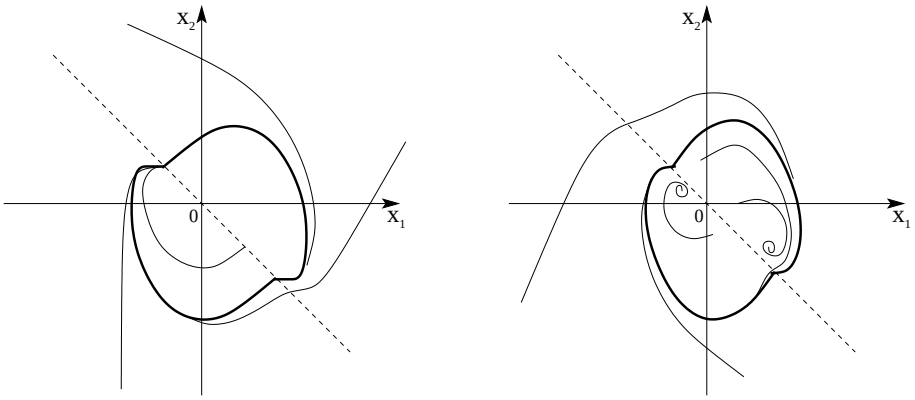
$$\dot{x}_2 = -2x_1 + x_2 - x_2\left(|x_1| + |x_2|\right) + u, \qquad (\mathbf{35})_{\text{ix}}$$

where u is the input to the system and x_1 and x_2 the state variables. Because the simultaneous equations $x_1 + x_2 - x_1\left(|x_1| + |x_2|\right) = 0$ and $-2x_1 + x_2 - x_2\left(|x_1| + |x_2|\right) = 0$ are ill-formed, there can be no equilibrium points.



In Figuer 9.1 is shown the phase plane of the system described by Equations 9.34 and 9.35 when there is no input. The limit cycle in Figure 9.1 is elliptical with the major axis along the x_2-axis. When the control input is $u = \text{sgn}(x_1 + x_2)$ the phase plane looks like Figure 9.2 (a) and when $u = -(x_1+x_2)+\text{sgn}(x_1+x_2)$ Figure 9.2 (b).

Figure 9.1 Phase plane of Equations 9.34 and 9.35 with no input.



(a) (b)

When there is no control Figure 9.1 shows that the equilibrium point at the origin is an unstable focus and there is a limit cycle circling around the origin. This limit cycle is slightly larger in the x_2 direction than in the x_1 direction. Every trajectory starting from an initial point other than the origin goes to and then stays on this limit cycle. With the control input $u = \text{sgn}(x_1 + x_2)$ there is a discontinuity on the surface of the hyperplane $s = x_1 + x_2 = 0$. All trajectories still converge to the limit cycle although the latter is distorted where it intersects the hyperplane. There is no node and every point on the hyperplane and inside the limit cycle is an unstable node. All these also hold when the control input is $u = -(x_1 + x_2) + \text{sgn}(x_1 + x_2)$ and there are two additional nodes as shown in Figure 9.2 (b).

During September and October 1996 I gave a series of seminars on a topic related to polytopes of polynomials. The topic I chose was recent (Pujara, 1996) and rather difficult for me but for me it was a success because the methods introduced was discussed weekly for at least a month in a series of the subsequent seminars, and they resulted in the idea being successfully applied, in the context of Pulse Width Modulation, by one of the students who attended in his Ph.D. work. The following is a recapture of the original seminar I gave. The discussions, which was the most interesting part in these seminars, are lost. A pseudoboundary is defined to be the set of all polynomials in the polytope each of which has at least one zero on the imaginary axis. A section of the pseudoboundary corresponding to ω_0 is a polytope whose vertices lie in the exposed 2-d faces of the given polytope. Pujara (1996) in his study of the stability boundary problem gave an algorithm to generate all the vertex polynomials of any section of the pseudoboundary of a polytope. A polytope is stable if and only if its exposed edges are stable. Interval polynomials are a hyperrectangle in coefficient space with edges parallel to the coordinate axis. Kharitonov's results is that if a polynomial family consists of interval polynomials, then the stability of just four extreme polynomials is both necessary and sufficient for the stability of the entire rectangle. Consider the polynomial $f(s,q) = s^n + a_1(q)s^{n-1} + a_2(q)s^{n-2} + \ldots + a_{n-1}(q)s + a_n(q)$ which produces an r-dimensional polytope P in R^{n+1}. Here $a_i(q)$, $i = 1$ to n, are real affine coefficients and $q_i^- \leq q_i \leq q_i^+$ for every q_i, $1 \leq i \leq r$. For fixed q, this polynomial is a point in R^{n+1} whose coordinates are the coefficients of the polynomials. The pseudoboundary, β, is the set of polynomials in the polytope P each of which has at least one zero on the imaginary axis. The section, β_0, of β at ω_0 consists of those polynomials each of which has a zero at $j\omega_0$. If $Z \in C^{n+1}$ is a set of all zeros of this polynomial, then $\omega_0 \in W$ if and only if $\exists g(j\omega_0) \in P$ such that $g(j\omega_0) = 0$. Every $\beta(j\omega_0)$ is a polytope which has its vertices on the exposed 2-d faces of P, which in turn can be explicitely determined.

Theorem. *(Pujara, 1996) The section of the pseudoboundary β of a polytope P at any frequency ω_0 is a polytope. The vertices of this pseudoboundary lie in the exposed 2-d faces of the polytope P.*

The state equations of a dc motor are

$$J\frac{d\omega}{dt} = ki \qquad (\mathbf{36})_{\text{ix}}$$

$$L\frac{di}{dt} = -k\omega - Ri + u \qquad (\mathbf{37})_{\text{ix}},$$

where i is the armagure current, u the voltage, R the resistance, L the inductance, J the moment of inertia, and ω the angular speed. The constant excitation flux ϕ results in the torque ki and the back e.m.f. $k\omega$.

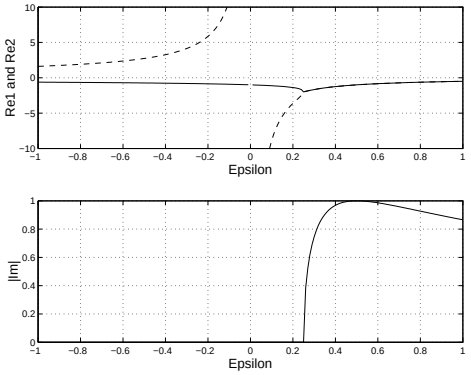


Figure 9.4 The eigenvalues of the dc motor as a function of ε..

My technical report number 6 (1998) I studied the control of synchronous machines. In the introduction I wrote to Professor Furuta to say that he may use any of the material without acknowledging me. I see it fits to describe briefly here some of the results because I initiated and carried out the work, as is normally the case with most of my researches done while in Japan. The electrical frequency of such machines synchronises with the mechanical speed. However it is more convenient to express angles in electrical unit, *i.e.* electrical angle, rather than in mechanical unit, *i.e.* mechanical angle, for the reason that a synchronous machine can have more than two poles. Whichever is the case, one may consider only a single pair of poles and then consider the electrical, mechanical, and magnetic conditions, which are associated with all the other pole pairs as repetitions of those for the pair being considered (*cf* Fitzgerald *et al*, 1971).

The electrical and the mechanical angles of a generator are related by the relation $\theta_e = \frac{p}{2}\theta_m$, where θ_e is the electrical angle of the output, θ_m is the angle of the rotating shaft, and p is the number of poles of the machine. The frequency of the voltage is $f = \frac{p}{2}\frac{n}{60}$ Hz, where n is the mechanical speed in rpm and $\frac{n}{60}$ the speed in revolution per second. The radian frequency ω of the voltage is $\omega_e = \frac{p}{2}\omega_m$, where ω_m is the mechanical speed in radian per second.

Figure 9.5 shows the dynamics of the mechanical and the electrical angles of a synchronous machine with three pair of poles. The solid line is the mechanical, while the dash-dot line the electrical rotation. The mechanical and the electrical rotation of a synchronous machine are juxtaposed in Figure 9.5. In power system engineering one wants to control the power and the frequency. Examples of controllers are speed govenors, actuators, regulators, and signal transducers. Speed govenors can be described by $\Delta f \cdot K = T_L$, where Δf is the change in frequency, K the gain, and T_L the load torque. The other controllers mentioned are high-gain power-amplif which convert things like position, oil pressure, and electricity into valve positions.

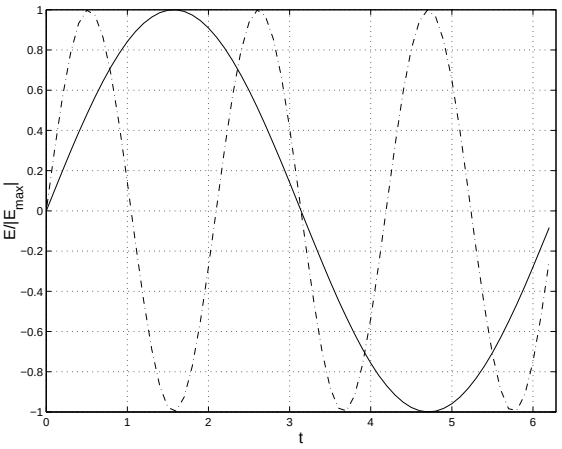


Figure 9.5 Dynamics of the mechanical and the electrical angles.

The change in load in the power-frequency control of a synchronous generator can either be a *load shedding* where the load decreases causing the frequency to increase, a *generator shedding* where the load increases causing the frequency to decrease, or a *short circuit* where the generator is suddenly disconnected and as a result the frequency increases to the maximum value. The effects caused by the load change are related to the equation $J\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = T_{PM} - T_E$, where J is the moment of inertia of the rotating mass, θ the angular position of the rotating mass, T_{PM} the prime-mover torque, and T_E the electrical torque.

In 1998 I presented a paper (§ E.22) at the ATAC-98 conference in Japan. The objectives of that study are to study the effects of a sign function input, to study the effects of ε in a singularly perturbed system, and to study a variable structure singularly perturbed system in other words those which have a sign function in the control input. *Singular perturbation* of any system implies that it has ε_i terms multiplying the term $\dot{\mathbf{x}}$ and that these terms ε_i approach zero. One can write down a given transfer function $T(s)$ in the state space forms $\dot{\mathbf{x}} = \mathbf{E}^{-1}\mathbf{A}\mathbf{x} + \mathbf{E}^{-1}\mathbf{B}\mathbf{u}$ and $y = \mathbf{C}\mathbf{x} + D$. For example, $T(s) = \frac{2(s+5)(s^2+4s+29)}{(s+10)(s+1)(s^2+8s+20)}$ can be represented by the above state equations with

$$\mathbf{A} = \begin{bmatrix} -8 & -5 & 1 & 1.25 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & -11 & -2.5 \\ 0 & 0 & 4 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 2.82843 \\ 0 \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{C} = [-2.83 \quad 1.6 \quad 0.71 \quad 0.88], \quad \text{and} \quad D = 0$$

Then if we rewrite $\mathbf{E}$ as

$$\mathbf{E} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \varepsilon \end{bmatrix},$$

we can study the perturbation effect of the singularly in ε as it decreases from $\varepsilon < 1$ towards $\varepsilon \ll 1$, namely

$$\begin{aligned}
\varepsilon = 0.5, T(s) &= \frac{2(s+10)(s^2+4s+29)}{(s^2+8s+20)(s+2.2984)(s+8.7015)} \\
\varepsilon = 0.1, T(s) &= \frac{2s^3+108s^2+458s+2900}{(s^2+8s+20)(s^2+11s+100)} \\
\varepsilon = 0.01, T(s) &= \frac{2s^3+1008s^2+4058s+29000}{(s^2+8s+20)(s^2+11s+1000)} \\
\varepsilon = 0.001, T(s) &= \frac{2s^3+10008s^2+40058s+(2.9\times 10^5 2.9)}{s^4+19s^3+10108s^2+80220s+(2\times 10^5)} \\
\varepsilon = 0.0001, T(s) &= \frac{2s^3+100008s^2+400058s+(2.9\times 10^6)}{s^4+19s^3+(1.0011\times 10^5)s^2+(8.0022\times 10^5)s+(2\times 10^6)}.
\end{aligned}$$

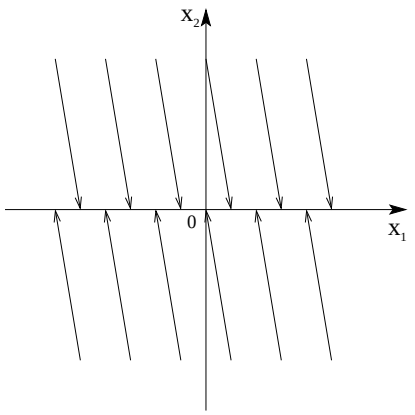
To summarise the root locus behaviour when ε decreases, two complex zeros are fixed, that is $s^2+4s+29$. Two complex poles described by $s^2+8s+20$ remain the same throughout. The zero on the real axis moves left. The two poles on the real axis moves toward each other. At the point when the moving zero passes the pole on its left hand side the root locus changes its characteristics.

Even as early as November 1998 I have decided that Control Systems will not be the only area that I will do researches in, as can be seen in the title of the technical report number two (1998) which has the words *system and control* put in brackets. Here among other things I considered the system described by the differential equation

$$\varepsilon\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \frac{\mathrm{d}x}{\mathrm{d}t} = u \qquad (\mathbf{38})_{\mathrm{ix}}$$

. This can be rewritten as the state equations $\dot{x}_1 = x_2$ and $\varepsilon\dot{x}_2 = -x_2 + u$ which represent a model in the *standard form* since the second equation has a real isolated root when $\varepsilon = 0$. By letting the right hand side of the state equation be zero, every point on the x_1 axis is an equilibrium point. A model described by the equations $\dot{x} = f(t,x,z,\varepsilon)$, $x \in R^n$ and $\varepsilon\dot{z} = g(t,x,z,\varepsilon)$, $z \in R^n$ is said to be in the standard form if and only if $0 = g(t,x,z,0)$ has k isolated real roots $z = h_i(t,x)$, $i = 1,2,\ldots,k$ (Khalil, 1996); then these two equations reduces to a *quasi-steady state* or *slow model* $\dot{x} = f(t,x,h(t,x),0)$.

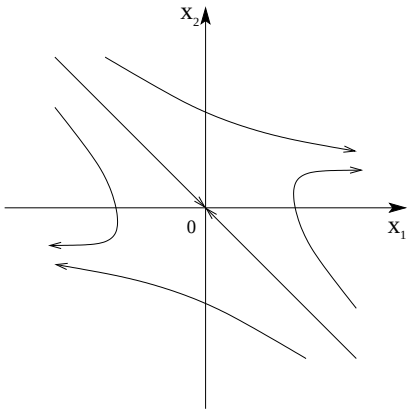
For the present system the eigenvalues obtained from the Jacobian matrix $A = \frac{\partial f}{\partial x}\Big|_{x=(x_1,0)} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix}$ are 0 and $-\frac{1}{\varepsilon}$. Therefore it has an equilibrium subspace and the qualitative behaviour of the trajectories depends on the values of ε, that is when $\varepsilon > 0$ all trajectories converge to the equilibrium subspace, when $\varepsilon = 0$ the system degenerates and is reduced to a first order system, and when $\varepsilon < 0$ then all trajectories diverge from the equilibrium subspace. In other words the system is stable when $\varepsilon > 0$ and unstable when $\varepsilon < 0$. By plotting the state planes and the time response of the state variables of this system when $u = 0$, one can see that the system is stable if $\varepsilon > 0$, and unstable if $\varepsilon < 0$. When $|\varepsilon|$ becomes smaller the response of the system becomes more rapid as $\dot{x}_2 = \frac{1}{\varepsilon} x_2$ rapidly converges to its root $x_2 = 0$ when ε approaches zero. The trajectory is a straight line which becomes more parallel to the x_2-axis the closer ε gets to zero.



As ε approaches zero and $u = 0$ the system becomes $x_2 = 0$ or $\dot{x}_1 = 0$ which has the solution $x_1(t) = x_1(0)$ and $x_2(t) = 0$. The singular point $\varepsilon = 0$ is the point of discontinuity since the solution is stable when $\varepsilon \to 0^+$ and unstable when $\varepsilon \to 0^-$.

Figure 9.6 State plane of the system $\varepsilon \mathrm{d}^2x/\mathrm{d}t^2 + \mathrm{d}x/\mathrm{d}t = u$ with $0 < \varepsilon < 1$..

Next study the effect of a feed back with the sign function, $u = K\,\mathrm{sgn}(Cx_1 + x_2)$, to the system in Equation 9.38, where C and K are real constants. When $C = K = 1$, simulation results showed that the response of the controlled system changes more rapidly as $|\varepsilon|$ becomes smaller. In other words, the response becomes faster as ε approaches zero. This increase in speed of the response as ε decreases in magnitude applies for the unstable case where $\varepsilon < 0$ as well as when $\varepsilon > 0$.



The boundary layer or the transient period of the system remains the same either with or without the input, that is t of approximately $[0, 0.05]$ and $[0, 0.4]$ for $\varepsilon = 0.01$ and 0.1 respectively. For $\varepsilon \geq 1$ this boundary layer extends beyond one second. With the feed back input, the state plane trajectory reaches the line $x_2 = 0$ during the transient period and then gently slides along it.

Figure 9.7 State plane of $\varepsilon \mathrm{d}^2x/\mathrm{d}t^2 + \mathrm{d}x/\mathrm{d}t = \mathrm{sgn}(x_1 + x_2)$ when $\varepsilon \geq 1$..

The state equations with the input become $\dot{x}_1 = x_2$ and $\varepsilon\dot{x}_2 = -x_2 + \mathrm{sgn}(x_1 + x_2)$. When $\varepsilon = 0$ the latter equation above becomes $0 = -x_2 + \mathrm{sgn}(x_1 + x_2)$ which gives the root $\bar{x}_2 = \mathrm{sgn}(\bar{x}_1 + \bar{x}_2)$ that can not be written in the form $\bar{x}_2 = h(t, \bar{x}_1)$. This means that the equation has no isolated root and consequently one can not isolate the fast mode from the slow mode by introducing a new variable $y = x_2 - h(t, x_1)$. But for the purpose of finding the boundary layer model, let $h(t, x_1, x_2) = \mathrm{sgn}(x_1 + x_2)$. Then $y = x_2 - h(t, x_1, x_2) = x_2 - \mathrm{sgn}(x_1 + x_2)$ and as a consequence $x_2 = y + \mathrm{sgn}(x_1 + x_2)$. Introduce a new time variable τ obtained from $\varepsilon\frac{\mathrm{d}y}{\mathrm{d}t} = \frac{\mathrm{d}y}{\mathrm{d}\tau}$ or $\frac{\mathrm{d}\tau}{\mathrm{d}t} = \frac{1}{\varepsilon}$ when $\tau_0 = 0$, that is $\tau = \tau_0 + \int_{t_0}^{t} t\mathrm{d}t = \frac{t - t_0}{\varepsilon}$. This new time variable τ is generally known as the *stretched* time variable. Then the equation $\varepsilon\left[\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\mathrm{d}}{\mathrm{d}t}\mathrm{sgn}(x_1 + x_2)\right] = -y$ becomes $\frac{\mathrm{d}y}{\mathrm{d}\tau} + \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{sgn}(x_1 + x_2) = -y$. Substituting $x_2 = y + \mathrm{sgn}(x_1 + x_2)$ gives $\frac{\mathrm{d}y}{\mathrm{d}\tau} = -y - \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{sgn}(x_1 + y + \mathrm{sgn}(x_1 + x_2))$, and therefore the boundary layer model is $\frac{\mathrm{d}y}{\mathrm{d}\tau} = -y - \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{sgn}(x_1 + x_2 + 1)$ when $x_1 + x_2 > 0$, is $\frac{\mathrm{d}y}{\mathrm{d}\tau} = -y - \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{sgn}(x_1 + x_2)$ when $x_1 + x_2 = 0$, or is $\frac{\mathrm{d}y}{\mathrm{d}\tau} = -y - \frac{\mathrm{d}}{\mathrm{d}\tau}\mathrm{sgn}(x_1 + x_2 - 1)$ when $x_1 + x_2 < 0$. When $\bar{x}_1 + \bar{x}_2 > 0$ then $\bar{x}_2 = +1$ and the reduced problem becomes $\dot{x}_1 = +1$. Likewise when $\bar{x}_1 + \bar{x}_2 = 0$ then $\bar{x}_2 = 0$ and $\dot{x}_1 = 0$, and when $\bar{x}_1 + \bar{x}_2 < 0$ then $\bar{x}_2 = -1$ and $\dot{x}_1 = -1$.

The discontinuous nature of the signum function makes the analysis of the variable structure control system difficult. This is due to the fact that one can not analytically find $\frac{\partial\,\mathrm{sgn}\,x(t)}{\partial t}$, $\frac{\partial\,\mathrm{sgn}\,x(t)}{\partial x_i}$, where $i = 1, 2, \ldots, n$ and $x(t)$ is an n-tuple vector. One way to overcome or go around this problem is to use a numerical approximation for the sign function where necessary.

Let the input be $u = -(x_1 + x_2) + \mathrm{sgn}(x_1 + x_2)$, then simulations show that the system has two timescales, which is the characteristic of a singularly perturbed system. Having a fast response within the boundary layer and a slow response elsewhere, the boundary layer becomes narrower and the response faster as ε approaches zero from above ($\varepsilon \to 0^+$). Also, numerical studies shows that this boundary layer, in seconds, is approximately $[0, 0.05]$, $[0, 0.23]$, and $[0, 0.54]$ respectively when ε is 0.01, 0.1, and 1. With this input, the equations of the system become $\dot{x}_1 = x$ and $\varepsilon\dot{x}_2 = -x_1 - 2x_2 + \mathrm{sgn}(x_1 + x_2)$. The last equation becomes $= 0 - x_1 - 2x_2 + \mathrm{sgn}(x_1 + x_2)$ when $\varepsilon = 0$, in other words $x_2 = -\frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}(x_1 + x_2)$. To find the boundary layer, change the variable x_2 to $y = x_2 - x_2|_{\varepsilon=0} = x_2 + \frac{1}{2}x_1 - \frac{1}{2}\mathrm{sgn}(x_1 + x_2)$. Therefore $x_2 = y - \frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}(x_1 + x_2) = y - \frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}\left(x_1 + y - \frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}(x_1 + x_2)\right)$. Which is essentially that x_2 equals $y - \frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}(y + \frac{1}{2}x_1 + \frac{1}{2})$ when $x_1 + x_2 > 0$, $y - \frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}(y + \frac{1}{2}x_1)$ when $x_1 + x_2 = 0$, and $y - \frac{1}{2}x_1 + \frac{1}{2}\mathrm{sgn}(y + \frac{1}{2}x_1 - \frac{1}{2})$ when $x_1 + x_2 < 0$. Substitute this into the equation for $\varepsilon\dot{x}_2$ above to get the boundary layer model $\varepsilon\left[\frac{\mathrm{d}y}{\mathrm{d}t} - \frac{1}{2}\frac{\mathrm{d}x_1}{\mathrm{d}t} + \frac{1}{2}\mathrm{sgn}(x_1 + x_2)\right] = -2y$ or $\varepsilon\left[\frac{\mathrm{d}y}{\mathrm{d}t} - \frac{y}{2} + \frac{1}{4}x_1 - \frac{1}{4}\mathrm{sgn}(x_1 + x_2) + \frac{1}{2}\mathrm{sgn}(x_1 + x_2)\right] = -2y$. Use the stretched time variable τ introduced above. Then $\frac{\mathrm{d}y}{\mathrm{d}\tau} = -\frac{3}{2}y - \frac{1}{4}x_1 - \frac{1}{4}\mathrm{sgn}(x_1 + x_2)$, which is the same as saying that $\frac{\mathrm{d}y}{\mathrm{d}\tau}$

equals $-\frac{3}{2}y - \frac{1}{4}x_1 - \frac{1}{4}$ when $x_1 + x_2 > 0$, $-\frac{3}{2}y - \frac{1}{4}x_1$ when $x_1 + x_2 = 0$, and $-\frac{3}{2}y - \frac{1}{4}x_1 + \frac{1}{4}$ when $x_1 + x_2 < 0$.

Now for a variable structure control, let the hyperplane be described by $s = cx_1 + x_2$ whose derivative is $\dot{s} = c\dot{x}_1 + xd_2$. Let the input be $u = -K \operatorname{sgn} s = -K \operatorname{sgn}(cx_1 + cx_2)$ for $K \in R^+$. Choose a Lyapunov function as $V = \frac{1}{2}s^2$. The state equations become $\dot{x}_1 = x_2$ and $\dot{x}_2 = -\frac{1}{\varepsilon}x_2 - \frac{K}{\varepsilon}(cx_1 + x_2)$. The derivative of the Lyapunov function is then $\dot{V} = s\dot{s} = (cx_1 + x_2)\left(cx_2 - \frac{1}{\varepsilon}x_2 - \frac{K}{\varepsilon}(cx_1 + x_2)\right) = c^2x_1x_2 - \frac{cx_1x_2}{\varepsilon} + cx_2^2 - \frac{x_2^2}{\varepsilon} - \frac{cKx_1}{\varepsilon}\operatorname{sgn}(cx_1 + x_2) - \frac{Kx_2}{\varepsilon}(cx_1 + x_2)$. In order to observe the effect of ε on the stability of the controlled system, let $c = 2.7$ and $K = 3.3$. From plots of $\dot{V}$ against x_1 and x_2 one may see that whether $\dot{V}$ be positive or negative depends upon the value ε takes. For example, there exists x which gives $\dot{V} > 0$ when ε is 1.7 and 0.08, while if $\varepsilon = 0.3$ then $\dot{V} < 0$ for all possible values of x.

The original system (Equation 9.38) has the eigenvalues at $\lambda_1 = 0$ and $\lambda_2 = -12.5$, with the corresponding eigenvectors at $v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $v_2 = \begin{bmatrix} -0.0797 \\ 0.9968 \end{bmatrix}$ respectively. The system is linear since A is a constant matrix and does not depend upon x. Also A is not a stability matrix since the condition $\operatorname{Re}\lambda_i < 0$ is not satisfied. The equilibrium point at the origin is not unique as it has already been pointed out earlier that every point on the x_1-axis is an equilibrium point. The x_1-axis is a nontrivial null space of the matrix A. The change of variables $z = M^{-1}x$, where $M = [v_1 \quad v_2] = \begin{bmatrix} 1 & -0.0797 \\ 0 & 0.9968 \end{bmatrix}$, leads to $M\dot{z} = AMz = \begin{bmatrix} 0 & 0 \\ 0 & -12.5 \end{bmatrix}$, and therefore $A = \begin{bmatrix} 0 & 0 \\ 0 & -12.5 \end{bmatrix}$. The solution to this system is $\dot{z}_1(t) = z_1(0)$ and $z_2(t) = z_2(0)e^{-12.5t} = z_2(0)e^{\lambda_2 t}$. That A is no stability matrix can be seen by trying to solve the Lyapunov equation $PA + A^TP = -Q$, where Q is a positive definite symmetric matrix. The Lyapunov equation for this system yields no solution.

From simulation results, when $\varepsilon < 0$ the controlled system is unstable, while for $\varepsilon > 0$ increasing K will widen the range of ε in which the controlled system is stable. That is, the larger the value of the gain K of the control input, the more robust with respect to ε this control scheme.

Note also that one way to stabilise the origin of the system is to solve the equation $A^TPE + E^TPA - (E^TPB + S)R^{-1}(B^TPE + S^T) + Q = 0$, or equivalently $F^TPE + E^TPF - E^TPBR^{-1}B^TPE + Q - SR^{-1}S^T = 0$, where $F = A - BR^{-1}S^T$, Q and R are symmetric positive definite matrices, and the system is written in the form $E\dot{x} = Ax + Bu$. For example, letting $Q = R = E = I$ and $S = 0$ leads to $P = \begin{bmatrix} 12.62 & 1 \\ 1 & 0/1194 \end{bmatrix}$. The feedback input is then $u = -Gx$, where $G = R^{-1}(B^TPE + S^T)$. The gain matrix will be $G = [1 \quad 0.1194]$. Then the closed loop eigenvalues can be computed from $(A - BG)V = EV\Gamma$, where $V = [v_1 \quad v_2]$, $\Gamma = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$, λ_i is the i^{th} eigenvalue, and v_i its corresponding eigenvector. In solving the Riccati equation the Frobenius norm of the relative residual matrix is 3.14×10^{-16}. Simulations on Simulink show that when $\varepsilon = 0$ the system is unstable. There exists a domain of attraction outside which the trajectory does not converge.

§ 9.3 TEXnicalities

I started to use Latex in 1994 to write a design exercise report for Dr. Martin Zarrop of the Control System Centre at UMIST. Back then, most people I know liked the program better than its progenitor TEX; even a computer expert like Dr. J. Braham Levy thought TEX difficult. I used Latex also for writing up my master degree dissertation in 1995, and continued using while in Japan and even as late as in early 2001. Since Latex runs on TEX but keeps creating new versions, one after another, the various version not being always compatible with one another, I decided to give the latter a try and have since completely switched over to it. In a way, Latex is a macro which runs on TEX, but it is more like a black box than a macro because one does not have the code for it, neither can one sufficiently tailor-make anything.Moreover, it greatly reduces the ability of a user to write his own macros by disabling many of such ability inherent in TEX. The first record I have of my using TEX is the source file of one of my Work Notes, dated 12^{th} February 2001 .

LaTEX(Lamport, 1985) is written by Leslie Lamport. Newer versions of it has come up at a regular interval. Unlike most other macros which run on TEX, her source code is free for none but herself. Moreover, having used TEX to do what it wants, LaTEX thereby castrates her progenitor in such a way that it is impossible for her users to define new macros efficiently by using the `\def` command. With `\def` disabled, the lion has lost its fangs and users become like lobotomised patients, dependent and docile. I have no doubt that with some of the TEX users having such idea as mine, sooner or later LaTEX will have to change in these respects. But this is the way I see the situation at the moment.

One of the first macros that I have written is the code to change the date format. The algorithm first sets $x = date$, then it assigns the ordinate endings *st, nd, rd* or *th* depending on the value of *ordinate*, which is calculated from

if $x > 30$ then $ordinate = 1$ else
  if $x > 20$ then $ordinate = x - 20$ else
    $ordinate = date$
  endif
endif

In the past when I still used LaTEX I could hardly adjust, but has to accept the existing formats. One awkward example is seen in my technical report number 6 where one reference (Fitzgerald *et al*, 1971) appears everywhere as "[FCKK71]", which looks very awkward.

The macros which I either wrote myself or adapted from elsewhere, mainly from the *manmac* macroby Knuth, are listed in § 9.3. Apart from these, this thesis uses the *plain* and the *manmac* macros. Another set of macros which I devellop is that which contains languages. As my definition of language is quite wide, this contains many languages among which are those which are used here, for example the lanugages for Chemistry, Chinese, Czech, German, French, grammatical jargons, Japanese, Lanna, Latin, latin grammar, Mathematics, Pali, Physics, Russian, Sanskrit, and Thai (Daiy), *etc*. But since Graham Davies says that no applications in language may be included in this thesis, these macros together with their explanation will be published elsewhere.

When I first started using TEX instead of LaTEX, I only used the macro `plain.tex`. Then in my first book typeset with TEX (Tiyapan, 2001, KNT2(ii)), I used in addition to the plain TEX `manmac.tex` and `epsf.tex`. Now to my amazement, I have discovered many other excellent macros, for instance `rotate.tex`, and found that I could understand how they work when I read them. This is one of the benefits that comes with talking in TEX instead of, for instance, LaTEX. I also know now the difference between the primitive TEX and plain TEX, and that the latter is only one of the infinitely many possible implementations of TEX. However, since all TEX gurus I know use plain TEX as a basis, there is no reason why I should be too proud to follow the practice. Having said that, my next plan is to improvise on the primitive TEX without any direct reference to the plain TEX macro.

The citation program BibTEX was intended to be used with LaTEX. Karl Berry and Oren Patashnik have written `btxmac.tex` which makes BibTEX usable from plain TEX. But for the present purpose I merely use my own macros, which are much simpler, and do not need BibTEX. Ultimately such database program as BibTEX would have been extremely useful. But I wish to develop something similar to it on my own.

10

§ 10. Conclusion

To summarise, I have began my study from the Voronoi network, and then went on to study continua and then tried to combine them together. I have suggested the idea that a percolation in continua can be represented by a percolation in a lattices the type of which depends on the attrition mechanism of the particles in that continuum, in other words the way they pack together.

I have written programs, and have listed a full half of them, and that is quite a lot in any programer's standard. This is because I try to write everything myself, and not to copy even when it comes to taking codes from programs that I have written myself in the past. I try to do it this way whenever possible, and as a result the various programs should have at least a few things out of common with each other.

Eventhough percolation already models a great many natural phenomena, I think it will prove to underly the fundamental structure of a great many more, even of the Big Bang. Therefore I truely believe that a better percolation theory may be what is behind the working of the universe.

A

§ A. Programs

§ A.1 Object location

```
% Object2dSqua.m
clear all; figure(1); clf; a=1; b=2; c=3;
a1=1; a2=1; c1=10; c2=10; b1=-2*a1*5; b2=-2*a2*7; P=eye(6); lambda=.98;
DithVar=.1; NoiseVar=.1; Limit=10; N=1000; x=2; y=3;
[Object,m] =ObjSqu(x,y);
subplot(2,2,1),Draw(Object,m,Limit,'Object','c');
xlabel('X'),ylabel('Y'); axis equal; axis([-10 10 -10 10]);
subplot(2,2,2),Layout(Object,Limit,m);
Real=RealLin(x,y,N); zeta=[a1,a2,b1,b2,c1,c2]'; t=0; s2=0; s3=0;
for Steps=1:N
  v =dither(DithVar); p =-b1/a1/2 +v; q =-b2/a2/2 +v;
  [Image,n] =ImgMtx(round(p),round(q));
  if Steps==10
    Move1(:,1) =Image(:,1); Move1(:,2) =Image(:,2);
  elseif Steps==100
    Move2(:,1) =Image(:,1); Move2(:,2) =Image(:,2);
  elseif Steps==200
    Move3(:,1) =Image(:,1); Move3(:,2) =Image(:,2);
  elseif Steps==300
    Move4(:,1) =Image(:,1); Move4(:,2) =Image(:,2);
  elseif Steps==400
    Move5(:,1) =Image(:,1); Move5(:,2) =Image(:,2);
  elseif Steps==500
    Move6(:,1) =Image(:,1); Move6(:,2) =Image(:,2);
  elseif Steps==600
    Move7(:,1) =Image(:,1); Move7(:,2) =Image(:,2);
  elseif Steps==700
    Move8(:,1) =Image(:,1); Move8(:,2) =Image(:,2);
  elseif Steps==800
    Move9(:,1) =Image(:,1); Move9(:,2) =Image(:,2);
  end;
  g =Compare(Object,Image,m,n); e =noise(NoiseVar); y =g+e;
  if y>1
    x =[p*p,q*q,p,q,1,1]'; t =t+1; Est(Steps,1) =t;
    Est(Steps,2) =-b1/a1/2; s2=Est(Steps,2);
    Est(Steps,3) =-b2/a2/2; s3=Est(Steps,3);
    P =1/lambda*P*(eye(6)-(x*x'*P)/(lambda+x'*P*x));
    zeta =zeta+P*x*(y-x'*zeta); a1 =zeta(1,:); a2 =zeta(2,:);
    b1 =zeta(3,:); b2 =zeta(4,:);
  else
    t =t+1; Est(Steps,1) =t; Est(Steps,2) =s2; Est(Steps,3) =s3;
    p =RandSrch(Limit); q =RandSrch(Limit); b1 =-2*a1*p; b2 =-2*a2*q;
  end;
end;
Title1 ='X'; Title2 ='Y'; Title3 ='Phase Plane'; Title4 ='Object and Image';
XLab ='Steps'; YLab ='Mag';
subplot(2,2,3),compare2([Est(:,1),Est(:,2)],[Real(:,1),Real(:,2)],Title1,XLab,YLab)
subplot(2,2,4),compare2([Est(:,1),Est(:,3)],[Real(:,1),Real(:,3)],Title2,XLab,YLab)
figure(2); clf;
subplot(3,3,1),Draw(Object,m,Limit,'Object','r'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,2),Draw(Move1,n,Limit,'N = 10','y'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,3),Draw(Move2,n,Limit,'N = 100','y'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,4),Draw(Move3,n,Limit,'N = 200','y'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,5),Draw(Move4,n,Limit,'N = 300','y'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,6),Draw(Move5,n,Limit,'N = 400','y'); axis equal; axis([-10 10 -10 10]);
```

```
subplot(3,3,7),Draw(Move6,n,Limit,'N = 500','y'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,8),Draw(Move7,n,Limit,'N = 600','y'); axis equal; axis([-10 10 -10 10]);
subplot(3,3,9),Draw(Move8,n,Limit,'N = 700','y'); axis equal; axis([-10 10 -10 10]);
 % Compare.m
function [y] =Compare(Object,Image,m,n)
y =0;
for i=1:m
  for j=1:n
    if (Object(i,1)==Image(j,1))&(Object(i,2)==Image(j,2))
      y =y +1;
    end;
  end;
end;
 % compare2.m
function [] =compare2(data1,data2,graph_title,x_label,y_label)
plot(data1(:,1),data1(:,2),'-',data2(:,1),data2(:,2),'-');
grid on; title(graph_title); xlabel(x_label); ylabel(y_label);
 % dither.m
function [dither]=dither(dith_variance)
dith_limit=sqrt(3*dith_variance); dither=dith_limit*2*(.5-rand(1));
 % Draw.m
function []=Draw(Pict,Dim,Limit,Title,Colour)
for i=1:Dim
fill([(Pict(i,1)-.5),(Pict(i,1)+.5),(Pict(i,1)+.5),(Pict(i,1)-.5)],...
  [(Pict(i,2)-.5),(Pict(i,2)-.5),(Pict(i,2)+.5),(Pict(i,2)+.5)],Colour),hold on;
end;
axis([-Limit Limit -Limit Limit]); grid; title(Title);
 % ImgMtx.m
function [yy,n]=ImgMtx(p,q)
i=1;
for x=(p-1):(p+1)
  for y=(q-1):(q+1)
    yy(i,1)=x; yy(i,2)=y;i=i+1;
  end;
end;
n =size(yy,1);
 % Layout.m
function []=Layout(Object,Limit,m)
for i=-Limit:Limit
  x =i+Limit+1;
  for j=-Limit:Limit
    y =j+Limit+1; [Image,n] =ObjMtx(i,j); xx(x,1) =i; yy(y,1) =j;
    z(x,y) =Compare(Object,Image,m,n);
  end;
end;
mesh(xx,yy,z); title('Object Profile'); xlabel('x'); ylabel('y');
zlabel('Pixels overlapped'); grid;
 % noise.m
function [e]=noise(noise_sig)
e=sqrt(noise_sig)*randn(1);
 % ObjSqu.m
function [yy,n]=ObjSqu(p,q)
i=1;
for x=(p-1):(p+1)
  for y=(q-1):(q+1)
    yy(i,1) =x; yy(i,2) =y; i =i+1;
  end;
end;
n =size(yy,1);
 % RealLin.m
function [Real] =RealLin(x,y,N)
for i=1:N
  Real(i,1) =i; Real(i,2) =x; Real(i,3) =y;
end;
% RandSrch.m
function [y]=RandSrch(limit)
y =limit*2*(.5-rand(1));
```

§ A.2 Network percolation, two dimensions

```
% perco
clear all; St=sum(100*clock); rand('state',St); CNa=200; Dim=2; X=rand(CNa,Dim);
[Va,Ca]=voronoin(X); T=delaunayn(X); TN=size(T,1); VNa=size(Va,1); LB=0.05;
UB=0.95; IXa=zeros(VNa,1);  V=[]; Count=0; VCNa=[];
for i=1:CNa,
  VCNa=[VCNa;size(Ca{i},2)];
end
for i=1:VNa,
  if((Va(i,1)>LB & Va(i,1)<UB) & (Va(i,2)>LB & Va(i,2)<UB))
    V=[V;Va(i,:)]; Count=Count+1; IXa(i,1)=Count;
  end
end
VN=size(V,1); VCN=[]; Count=0; Xa=X; X=[];
Tmp=sparse(1,CNa);
for i=1:CNa,
  Include=1;
  for j=1:VCNa(i,1),
    if(IXa(Ca{i}(1,j),1)==0)
      Include=0;
    end
  end
  if(Include==1)
    Count=Count+1; C{Count,1}=[]; VCN=[VCN;VCNa(i,1)];
    for j=1:VCNa(i,1),
      C{Count,1}(1,j)=IXa(Ca{i}(1,j),1);
    end
    X=[X;Xa(i,:)]; Tmp(1,i)=Count;
  end
end
CN=size(C,1); T2=[]; T3=[];
for i=1:TN,
  TmpA=[];
  for j=1:3,
    if(Tmp(T(i,j)))
      TmpA=[TmpA,Tmp(T(i,j))];
    end
  end
  TmpB=size(TmpA,2);
  if(TmpB==2)
    T2=[T2;TmpA];
  elseif(TmpB==3)
    T3=[T3;TmpA];
  end
end
% for cells
B=[]; BXX=sparse(CN,CN); NeCMat=sparse(CN,CN); Count=0;
for i=1:size(T2,1),
  Count=Count+1; B=[B;[T2(i,1),T2(i,2)]]; BXX(T2(i,1),T2(i,2))=Count;
  BXX(T2(i,2),T2(i,1))=Count; NeCMat(T2(i,1),T2(i,2))=1; NeCMat(T2(i,2),T2(i,1))=1;
end
for i=1:size(T3,1),
  for j=1:Dim,
    for k=(j+1):(Dim+1),
      if(BXX(T3(i,j),T3(i,k))==0)
        Count=Count+1; B=[B;[T3(i,j),T3(i,k)]]; BXX(T3(i,j),T3(i,k))=Count;
        BXX(T3(i,k),T3(i,j))=Count; NeCMat(T3(i,j),T3(i,k))=1;
        NeCMat(T3(i,k),T3(i,j))=1;
      end
    end
  end
end
BN=Count; A=X; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N); LBc=0.2; UBc=1-LBc;
for i=1:N,
  if(A(i,1)<=LBc)
    LMat(1,i)=1;
  elseif(A(i,1)>=UBc)
    UMat(1,i)=1;
  end
end
NeMat=NeCMat; Blocked=randperm(CN);
% for bonds
NeBMat=sparse(BN,BN);
for i=1:CN,
  [a,b,c]=find(BXX(i,:)); nc=size(c,2);
  for j=1:(nc-1),
    for k=(j+1):nc,
      NeBMat(c(1,j),c(1,k))=1; NeBMat(c(1,k),c(1,j))=1;
    end
  end
```

```
end
A=B; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((X(A(i,1),1)<=LBc) | (X(A(i,2),1)<=LBc))
    LMat(1,i)=1;
  elseif((X(A(i,1),1)>=UBc) | (X(A(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=NeBMat; Blocked=randperm(BN);
%for vertices
Tmp=sparse(1,VN);
for i=1:CN,
  for j=1:VCN(i,1),
    Tmp(1,C{i}(1,j))=1;
  end
end
Vv=[];
Count=0;
for i=1:VN,
  if(Tmp(1,i))
    Count=Count+1; Vv=[Vv;V(i,:)]; Tmp(1,i)=Count;
  end
end
VvN=size(Vv,1);
for i=1:CN,
  for j=1:VCN(i,1),
    Cv{i}(1,j)=Tmp(1,C{i}(1,j));
  end
end
E=[]; EVV=sparse(VN,VN); EVVv=sparse(VvN,VvN); NeVMat=sparse(VvN,VvN);
Countv=0; Count=0;
for i=1:CN,
  Tmp=[Cv{i}(1,1:VCN(i,1)),Cv{i}(1,1)];
  for j=1:VCN(i,1),
    V1=Tmp(1,j); V2=Tmp(1,(j+1));
    if(NeVMat(V1,V2)==0)
      Countv=Countv+1; NeVMat(V1,V2)=1; NeVMat(V2,V1)=1;
    end
  end
  Tmp=[C{i}(1,1:VCN(i,1)),C{i}(1,1)];
  for j=1:VCN(i,1),
    V1=Tmp(1,j); V2=Tmp(1,(j+1));
    if(EVV(V1,V2)==0)
      Count=Count+1; E=[E;[V1,V2]]; EVV(V1,V2)=Count; EVV(V2,V1)=Count;
    end
  end
end
EN=Count; A=Vv; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
LBv=2*LB; UBv=(UB-LB);
for i=1:N,
  if(A(i,1)<=LBv)
    LMat(1,i)=1;
  elseif(A(i,1)>=UBv)
    UMat(1,i)=1;
  end
end
NeMat=NeVMat; Blocked=randperm(VvN);
%for edges
NeEMat=sparse(EN,EN); MEV=sparse(EN,VN); [a,b,c]=find(EVV); nc=size(c,1);
for i=1:nc,
  MEV(c(i),a(i))=1; MEV(c(i),b(i))=1;
end
for i=1:VN,
  a=find(MEV(:,i));
  if(~isempty(a))
    TmpN=size(a,1); Tmp=[a;a(1,1)]';
    for j=1:TmpN,
      for k=(j+1):TmpN,
        NeEMat(Tmp(1,j),Tmp(1,k))=1; NeEMat(Tmp(1,k),Tmp(1,j))=1;
      end
    end
  end
end
A=E; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((V(A(i,1),1)<=LBv) | (V(A(i,2),1)<=LBv))
    LMat(1,i)=1;
  elseif((V(A(i,1),1)>=UBv) | (V(A(i,2),1)>=UBv))
    UMat(1,i)=1;
  end
end
```

```
NeMat=NeEMat; Blocked=randperm(EN);
% perco1
clear ClusA ClusB TSeries; NClusA=0; Perco=0;
for i=1:N,
  Joined=0;
  for j=1:NClusA,
    if(ClusA{j,3}(1,Blocked(1,i))~=0)
      ClusA{j,1}=ClusA{j,1}+1; ClusA{j,2}(1,Blocked(1,i))=1;
      ClusA{j,3}=ClusA{j,3} | NeMat(Blocked(1,i),:); Joined=1;
    end
    if(Joined==1)
      for k=1:4,
        ClusB{1,k}=ClusA{j,k};
      end
      NClusB=1;
      if(j==1)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      elseif(j==NClusA)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
      else
        Tmp=ClusA; clear ClusA;
        for k=1:(j-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
        for k=j:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      end
      for k=1:(NClusA-1),
        if(sum(ClusA{k,2} & ClusB{1,3}) ~= 0)
          ClusB{1,1}=ClusB{1,1}+ClusA{k,1}; ClusB{1,2}=ClusB{1,2} | ClusA{k,2};
          ClusB{1,3}=ClusB{1,3} | ClusA{k,3}; ClusB{1,4}=ClusB{1,4} | ClusA{k,4};
        else
          NClusB=NClusB+1;
          for l=1:4,
            ClusB{NClusB,l}=ClusA{k,l};
          end
        end
      end
      if((sum(full(LMat & ClusB{1,2}))~=0) & (sum(full(UMat & ClusB{1,2}))~=0))
        ClusB{1,4}=1; Perco=1;
      end
      NClusA=NClusB; ClusA=ClusB; clear ClusB; break;
    end
  end
  if(Joined==0)
    NClusA=NClusA+1; ClusA{NClusA,1}=1;
    ClusA{NClusA,2}=sparse(1,Blocked(1,i),1,1,N);
    ClusA{NClusA,3}=NeMat(Blocked(1,i),:); ClusA{NClusA,4}=0;
  end
  TSeries{i,1}=ClusA; TSeries{i,2}=Perco;
end
% Reverse
Tmp=Blocked; Blocked=[];
for i=1:N,
  Blocked=[Blocked,Tmp(1,(N-i+1))];
end
% simulations
Nc=0; TSnap=[];
for i=1:N,
  if(TSeries{i,2})
    Nc=i; break;
  end
end
Pc=Nc/N; Cord=mean(sum(NeMat,2));
```

§ A.3 Network percolation, three dimensions

```
% perco3d.m
clear all; St=sum(100*clock); rand('state',St); CNa=300; Dim=3;
X=rand(CNa,Dim); [Va,Ca]=voronoin(X); T=delaunayn(X); TN=size(T,1);
VNa=size(Va,1); LB=0.05; UB=0.95; IXa=zeros(VNa,1); VCNa=[];
for i=1:CNa,
  VCNa=[VCNa;size(Ca{i},2)];
end
MVCa=[];
for i=1:CNa,
  Tmp=ones(1,VCNa(i,1)); MVCa=[MVCa;sparse(Tmp,Ca{i},Tmp,1,VNa)];
end
Vin=zeros(1,VNa); Count=0;
for i=1:VNa,
  if((max(Va(i,:))<1) & (min(Va(i,:))>0))
    Count=Count+1; Vin(1,i)=1; IXa(i,1)=Count;
  end
end
Tmp=~Vin; Cin=ones(1,CNa);
for i=1:CNa,
  if(sum(Tmp & MVCa(i,:)))
    Cin(1,i)=0;
  end
end
C=[]; count=0; VCN=[];
for i=1:CNa,
  if(Cin(i))
    count=count+1; TmpN=size(Ca{i},2);
    for j=1:TmpN,
      C{count,1}(1,j)=IXa(Ca{i}(1,j));
    end
    VCN(count,1)=TmpN;
  end
end
CN=size(C,1); MidBCx=sparse(CNa,CNa); MidBCy=sparse(CNa,CNa);
MidBCz=sparse(CNa,CNa); BLng=sparse(CNa,CNa);
for i=1:TN,
  Tmp=[T(i,:),T(i,1)];
  for j=1:(Dim+1),
    for k=(j+1):(Dim+1),
      if((Cin(1,Tmp(1,j)) | Cin(1,Tmp(1,k))) & ~BLng(j,k))
        MidBCx(Tmp(1,j),Tmp(1,k))=(X(k,1)+X(j,1))/2;
        MidBCx(Tmp(1,k),Tmp(1,j))=(X(k,1)+X(j,1))/2;
        MidBCy(Tmp(1,j),Tmp(1,k))=(X(k,2)+X(j,2))/2;
        MidBCy(Tmp(1,k),Tmp(1,j))=(X(k,2)+X(j,2))/2;
        MidBCz(Tmp(1,j),Tmp(1,k))=(X(k,3)+X(j,3))/2;
        MidBCz(Tmp(1,k),Tmp(1,j))=(X(k,3)+X(j,3))/2;
        dx=X(k,1)-X(j,1); dy=X(k,2)-X(j,2); dz=X(k,3)-X(j,3);
        TmpA=sqrt(dx*dx + dy*dy + dz*dz);
        BLng(Tmp(1,j),Tmp(1,k))=TmpA; BLng(Tmp(1,k),Tmp(1,j))=TmpA;
      end
    end
  end
end
Fa=[]; Count=0; FaC=[];
for i=1:CNa,
  if(Cin(1,i))
    FaC{i,1}=0; FaC{i,2}=[];
  end
end
for i=1:(CNa-1),
  for j=(i+1):CNa,
    TmpA=0; TmpB=0;
    if(Cin(1,i))
      TmpA=1;
    end
    if(Cin(1,j))
      TmpB=1;
    end
    if(TmpA | TmpB)
      Tmp=MVCa(i,:) & MVCa(j,:);
      if(sum(Tmp))
        [a,b]=find(Tmp); Count=Count+1; Fa{Count,1}=size(b,2); Fa{Count,2}=b;
        Fa{Count,3}=[MidBCx(i,j),MidBCy(i,j),MidBCz(i,j)];
        if(TmpA)
          FaC{i,1}=FaC{i,1} + 1; FaC{i,2}=[FaC{i,2},Count];
          FaC{i,3}{1,1}=i; FaC{i,3}{1,2}=j;
        end
        if(TmpB)
          FaC{j,1}=FaC{j,1} + 1; FaC{j,2}=[FaC{j,2},Count];
```

```
80            FaC{j,3}{1,1}=i; FaC{j,3}{1,2}=j;
81          end
82        end
83      end
84    end
85  end
86  FaN=size(Fa,1); V=[];
87  for i=1:VNa,
88    if(Vin(1,i))
89      V=[V;[Va(i,:),i]];
90    end
91  end
92  VN=size(V,1); Tmp=sparse(VNa,1);
93  for i=1:VN,
94    Tmp(V(i,4),1)=i;
95  end
96  F=Fa;
97  for i=1:FaN,
98    for j=1:F{i,1},
99      F{i,2}(1,j)=Tmp(Fa{i,2}(1,j),1);
100   end
101 end
102 FN=FaN; FC=[];
103 count=0;
104 for i=1:CNa,
105   if(Cin(i))
106     count=count+1; FC{count,1}=FaC{i,1};
107     FC{count,2}=FaC{i,2}; FC{count,3}=FaC{i,3};
108   end
109 end
110 NghV=sparse(VN,VN); Tmp=F; TmpN=FN;
111 for i=1:TmpN,
112   TmpA=Tmp{i,2}; x=[]; y=[]; z=[]; TmpB=Tmp{i,1};
113   if(TmpB==3)
114     for j=1:2,
115       for k=(j+1):3,
116         NghV(TmpA(1,j),TmpA(1,k))=1; NghV(TmpA(1,k),TmpA(1,j))=1;
117       end
118     end
119   else
120     for j=1:TmpB,
121       x=[x;V(TmpA(1,j),1)]; y=[y;V(TmpA(1,j),2)]; z=[z;V(TmpA(1,j),3)];
122     end
123     a=y(1)*(z(2)-z(3))+y(2)*(z(3)-z(1))+y(3)*(z(1)-z(2));
124     b=z(1)*(x(2)-x(3))+z(2)*(x(3)-x(1))+z(3)*(x(1)-x(2));
125     c=x(1)*(y(2)-y(3))+x(2)*(y(3)-y(1))+x(3)*(y(1)-y(2));
126     K=1/sqrt(a*a + b*b + c*c); Th{1}=K*a; Th{2}=K*b; Th{3}=K*c; Max=0;
127     for j=1:3,
128       if(Th{j}<Max)
129         Max=Th{j}; jMax=j;
130       end
131     end
132     if(jMax==1)
133       p=y; q=z;
134     elseif(jMax==2)
135       p=x; q=z;
136     else
137       p=x; q=y;
138     end
139     t=delaunay(p,q);
140     for j=1:size(t,1),
141       for k=1:3,
142         t(j,k)=TmpA(1,t(j,k));
143       end
144     end
145     Nt=size(t,1); TmpC=sparse(VN,VN);
146     for j=1:Nt,
147       TmpT=[t(j,:),t(j,1)];
148       for k=1:3,
149         TmpD=sort([TmpT(1,k),TmpT(1,(k+1))]);
150         TmpC(TmpD(1,1),TmpD(1,2))=TmpC(TmpD(1,1),TmpD(1,2))+1;
151       end
152     end
153     [k,l,m]=find(TmpC);
154     for j=1:size(k,1),
155       if(m(j)==1)
156         NghV(k(j),l(j))=1; NghV(l(j),k(j))=1;
157       end
158     end
159   end
160 end
161 Fed=[];
```

```
for i=1:FN,
  for j=1:2,
    Fed{i,j}=F{i,j};
  end
end
for i=1:FN,
  Count=0; TmpN=Fed{i,1};
  if(TmpN>3)
    Tmp=Fed{i,2}; TmpA=Tmp(1,1); Tmp=Tmp(1,2:TmpN); TmpN=TmpN-1; Count=Count+1;
    while(TmpN)
      a=TmpA(1,Count); TmpB=[]; Found=0;
      for j=1:TmpN,
        TmpC=Tmp(1,j);
        if(NghV(a,TmpC) & ~Found)
          TmpA=[TmpA,TmpC]; Found=1;
        else
          TmpB=[TmpB,TmpC];
        end
      end
      Tmp=TmpB; TmpN=TmpN-1; Count=Count+1;
    end
    Fed{i,2}=TmpA;
  end
end
LB2=2*LB; UB2=(UB-LB);
% cells II
Tmp=ones(1,CNa);
for i=1:CNa,
  if((max(X(i,:))>UB) | (min(X(i,:))<LB))
    Tmp(1,i)=0;
  end
end
a=find(Tmp); TmpB=sparse(1,CNa); x=[];
for i=1:size(a,2),
  TmpB(1,a(1,i))=i; x(i,:)=X(a(i),:);
end
xn=size(x,1); TmpA=zeros(size(T)); TmpN=size(T,1);
for i=1:TmpN,
  for j=1:4,
    if((Tmp(1,T(i,j))))
      TmpA(i,j)=TmpB(1,T(i,j));
    end
  end
end
nghc=sparse(xn,xn);
for i=1:TmpN,
  [a,b,c]=find(TmpA(i,:)); TmpB=size(c,2);
  if(TmpB>1)
    for j=1:(TmpB-1),
      for k=(j+1):TmpB,
        nghc(c(1,j),c(1,k))=1; nghc(c(1,k),c(1,j))=1;
      end
    end
  end
end
A=x; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if(A(i,1)<=LB2)
    LMat(1,i)=1;
  elseif(A(i,1)>=UB2)
    UMat(1,i)=1;
  end
end
NeMat=nghc; Blocked=randperm(xn);
% bonds II
[a,b,c]=find(triu(nghc)); b=[a,b]; bn=size(b,1); Tmp=sparse(bn,xn);
for i=1:bn,
  Tmp(i,b(i,1))=1; Tmp(i,b(i,2))=1;
end
nghb=sparse(bn,bn);
for i=1:xn,
  a=find(Tmp(:,i));
  if(~isempty(a))
    TmpN=size(a,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        nghb(a(j,1),a(k,1))=1; nghb(a(k,1),a(j,1))=1;
      end
    end
  end
end
A=b; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
```

```
for i=1:N,
  if((x(A(i,1),1)<=LB2) | (x(A(i,2),1)<=LB2))
    LMat(1,i)=1;
  elseif((x(A(i,1),1)>=UB2) | (x(A(i,2),1)>=UB2))
    UMat(1,i)=1;
  end
end
NeMat=nghb; Blocked=randperm(N);
% vertices II
[a,b,c]=find((NghV)); Tmp=sparse(1,VN);
for i=1:size(a,1),
  Tmp(1,b(i,1))=1;
end
d=find(Tmp); TmpN=size(d,2);
for i=1:TmpN,
  Tmp(1,d(1,i))=i;
end
for i=1:size(a,1),
  a(i,1)=Tmp(1,a(i,1)); b(i,1)=Tmp(1,b(i,1));
end
nghv=sparse(a,b,c,TmpN,TmpN); TmpA=[];
for i=1:TmpN,
  TmpA(Tmp(1,d(i)),:)=V(i,1:3);
end
A=TmpA; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if(A(i,1)<=LB2)
    LMat(1,i)=1;
  elseif(A(i,1)>=UB2)
    UMat(1,i)=1;
  end
end
NeMat=nghv; Blocked=randperm(N);
% edges II
[a,b,c]=find(triu(NghV)); E=[a,b]; EN=size(E,1); Tmp=sparse(EN,VN);
for i=1:EN,
  Tmp(i,E(i,1))=1; Tmp(i,E(i,2))=1;
end
NghE=sparse(EN,EN);
for i=1:VN,
  a=find(Tmp(:,i));
  if(~isempty(a))
    TmpN=size(a,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        NghE(a(j,1),a(k,1))=1; NghE(a(k,1),a(j,1))=1;
      end
    end
  end
end
A=E; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((V(A(i,1),1)<=LB2) | (V(A(i,2),1)<=LB2))
    LMat(1,i)=1;
  elseif((V(A(i,1),1)>=UB2) | (V(A(i,2),1)>=UB2))
    UMat(1,i)=1;
  end
end
NeMat=NghE; Blocked=randperm(N);
% percolation
clear ClusA ClusB TSeries; NClusA=0; Perco=0;
for i=1:N,
  Joined=0;
  for j=1:NClusA,
    if(ClusA{j,3}(1,Blocked(1,i))~=0)
      ClusA{j,1}=ClusA{j,1}+1; ClusA{j,2}(1,Blocked(1,i))=1;
      ClusA{j,3}=ClusA{j,3} | NeMat(Blocked(1,i),:); Joined=1;
    end
    if(Joined==1)
      for k=1:4,
        ClusB{1,k}=ClusA{j,k};
      end
      NClusB=1;
      if(j==1)
        Tmp=ClusA; clear ClusA;
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      elseif(j==NClusA)
        Tmp=ClusA; clear ClusA;
```

```
        for k=1:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
      else
        Tmp=ClusA; clear ClusA;
        for k=1:(j-1),
          for l=1:4,
            ClusA{k,l}=Tmp{k,l};
          end
        end
        for k=j:(NClusA-1),
          for l=1:4,
            ClusA{k,l}=Tmp{(k+1),l};
          end
        end
      end
      for k=1:(NClusA-1),
        if(sum(ClusA{k,2} & ClusB{1,3}) ~= 0)
          ClusB{1,1}=ClusB{1,1}+ClusA{k,1}; ClusB{1,2}=ClusB{1,2} | ClusA{k,2};
          ClusB{1,3}=ClusB{1,3} | ClusA{k,3}; ClusB{1,4}=ClusB{1,4} | ClusA{k,4};
        else
          NClusB=NClusB+1;
          for l=1:4,
            ClusB{NClusB,l}=ClusA{k,l};
          end
        end
      end
      if((sum(full(LMat & ClusB{1,2}))~=0) & (sum(full(UMat & ClusB{1,2}))~=0))
        ClusB{1,4}=1; Perco=1;
      end
      NClusA=NClusB; ClusA=ClusB; clear ClusB; break;
    end
  end
  if(Joined==0)
    NClusA=NClusA+1; ClusA{NClusA,1}=1;
    ClusA{NClusA,2}=sparse(1,Blocked(1,i),1,1,N);
    ClusA{NClusA,3}=NeMat(Blocked(1,i),:); ClusA{NClusA,4}=0;
  end
  TSeries{i,1}=ClusA; TSeries{i,2}=Perco;
end
% Reverse
Tmp=Blocked; Blocked=[];
for i=1:N,
  Blocked=[Blocked,Tmp(1,(N-i+1))];
end
% simulations
Nc=0;
for i=1:N,
  if(TSeries{i,2})
    Nc=i; break;
  end
end
Pc=Nc/N; Cord=mean(sum(NeMat,2));
```

§ A.4 Network percolation, 2–d section

```
% section
MVC=[];
for i=1:CN,
  Tmp=ones(1,VCN(i,1)); MVC=[MVC;sparse(Tmp,C{i},Tmp,1,VN)];
end
CE=[];
for i=1:EN,
  Tmp=MVC(:,E(i,1)) & MVC(:,E(i,2));
  if(sum(Tmp))
    TmpA=find(Tmp)'; TmpN=size(TmpA,2); CE{i,1}=TmpN; CE{i,2}=TmpA;
  end
end
ie=[]; je=[]; ke=[];
for i=1:EN,
  ie(i,1)=V(E(i,2),1)-V(E(i,1),1); je(i,1)=V(E(i,2),2)-V(E(i,1),2);
  ke(i,1)=V(E(i,2),3)-V(E(i,1),3);
end
a=1; b=.01; cc=0; d=-.5; v=[]; vC=[]; count=0;
for i=1:EN,
  Tmp=(a*ie(i)+b*je(i)+cc*ke(i));
  if(Tmp)
    v1=E(i,1); x1=V(v1,1); y1=V(v1,2); z1=V(v1,3); TmpA=(a*x1+b*y1+cc*z1+d);
    v2=E(i,2); x2=V(v2,1); y2=V(v2,2); z2=V(v2,3); t=-TmpA/Tmp;
    if((t>=0) & (t<=1))
      x=x1+(x2-x1)*t; y=y1+(y2-y1)*t; z=z1+(z2-z1)*t; count=count+1;
      v(count,:)=[x,y,z]; vC{count,1}=CE{i,1}; vC{count,2}=CE{i,2};
    end
  else
    if(~TmpA)  % both nom and denom = 0
      count=count+1; v(count,:)=[x1,y1,z1]; vC{count,1}=CE{i,1};
      vC{count,2}=CE{i,2}; count=count+1; v(count,:)=[x2,y2,z2];
      vC{count,1}=CE{i,1}; vC{count,2}=CE{i,2};
    end
  end
end
vn=count; cC=sparse(CN,1); count=0;
for i=1:vn,
  for j=1:vC{i,1},
    if(~cC(vC{i,2}(j)))
      count=count+1; cC(vC{i,2}(j),1)=count;
    end
  end
end
cn=count; vc=vC;
for i=1:vn,
  for j=1:vc{i,1},
    vc{i,2}(1,j)=cC(vC{i,2}(j));
  end
end
c=[];
for i=1:cn,
  c{i,1}=0; c{i,2}=[];
end
for i=1:vn,
  for j=1:vc{i,1},
    c{vc{i,2}(j),1}=c{vc{i,2}(j),1}+1; c{vc{i,2}(j),2}=[c{vc{i,2}(j),2},i];
  end
end
for i=1:cn,
  Tmp=[];
  for j=1:c{i,1},
    Tmp=[Tmp;v(c{i,2}(j),:),c{i,2}(j)];
  end
  TmpA=min(Tmp,[],1); TmpB=max(Tmp,[],1); [TmpC,TmpD]=min(TmpB-TmpA);
  if(TmpD==1)
    TmpA=Tmp(:,2); TmpB=Tmp(:,3);
  elseif(TmpD==2)
    TmpA=Tmp(:,1); TmpB=Tmp(:,3);
  else
    TmpA=Tmp(:,1); TmpB=Tmp(:,2);
  end
  TmpC=delaunay(TmpA,TmpB); TmpN=size(TmpC,1);
  for j=1:TmpN,
    for k=1:3,
      TmpC(j,k)=Tmp(TmpC(j,k),4);
    end
  end
  TmpA=sparse(vn,vn);
  for j=1:TmpN,
```

```
    for k=1:2,
      for m=(k+1):3,
        TmpA(TmpC(j,k),TmpC(j,m))=TmpA(TmpC(j,k),TmpC(j,m))+1;
        TmpA(TmpC(j,m),TmpC(j,k))=TmpA(TmpC(j,m),TmpC(j,k))+1;
      end
    end
  end
  [x,y,z]=find(TmpA); TmpB=[]; TmpC=[];
  for j=1:size(x,1),
    if(z(j)==1)
      TmpB=[TmpB;x(j),y(j)]; TmpC(y(j),1)=1;
    end
  end
  TmpA=[];
  for j=1:size(TmpC,1),
    TmpA{j,1}=[];
  end
  for j=1:size(TmpB,1),
    TmpA{TmpB(j,1),1}=[TmpA{TmpB(j,1),1},TmpB(j,2)];
  end
  Tmp=Tmp(1,4); TmpB=Tmp;
  TmpC=sparse(Tmp,1,1,vn,1); count=c{i,1}-1;
  while(count>0),
    if(~(TmpC(TmpA{Tmp}(1),1)))
      Tmp=TmpA{Tmp}(1); TmpB=[TmpB,Tmp]; TmpC(Tmp,1)=1;
    else
      Tmp=TmpA{Tmp}(2); TmpB=[TmpB,Tmp]; TmpC(Tmp,1)=1;
    end
    count=count-1;
  end
  c{i,3}=TmpB;
end
for i=1:cn,
  Tmp=[0,0,0];
  for j=1:c{i,1},
    Tmp=Tmp+v(c{i,2}(j),:);
  end
  Tmp=Tmp/c{i,1}; c{i,4}=Tmp;
end
Tmp=sqrt(a*a+b*b+cc*cc); u=[a/Tmp,b/Tmp,cc/Tmp]; uzp=u; ux=[1,0,0];
Tmp=cross(u,ux); TmpA=sqrt(Tmp(1)*Tmp(1)+Tmp(2)*Tmp(2)+Tmp(3)*Tmp(3));
uyp=Tmp/TmpA; uxp=cross(uyp,uzp); R=[uxp,0;uyp,0;uzp,0;0,0,0,1];
vp=(R*[v';ones(1,vn)])'; vp=vp(:,1:2); ad=min(vp,[],1);
vp=vp-[ad(1)*ones(vn,1),ad(2)*ones(vn,1)];; cs=[];
for i=1:cn,
  Tmp=R*[c{i,4}';1]; Tmp=Tmp(1:2)'-ad; c{i,5}=Tmp; cs=[cs;Tmp];
end
LB=min(vp(:,1)); UB=max(vp(:,1)); Tmp=UB-LB; LBv=LB+0.1*Tmp;
UBv=UB-LBv; Tmp=min(cs(:,1)); LB=min(cs(:,1)); UB=max(cs(:,1));
Tmp=UB-LB; LBc=LB+0.1*Tmp; UBc=UB-LBc;
% cell
cvm=sparse(cn,vn);
for i=1:cn,
  for j=1:c{i,1},
    cvm(i,c{i,2}(j))=1;
  end
end
nghc=sparse(cn,cn);
for i=1:(cn-1),
  for j=(i+1):cn,
    Tmp=find(cvm(i,:) & cvm(j,:));
    if(~isempty(Tmp))
      TmpN=size(Tmp,2);
      if(TmpN>1)
        for k=1:TmpN,
          nghc(i,j)=1; nghc(j,i)=1;
        end
      end
    end
  end
end
N=cn; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:cn,
  if(cs(i,1)<=LBc)
    LMat(1,i)=1;
  end
  if(cs(i,1)>=UBc)
    UMat(1,i)=1;
  end
end
NeMat=nghc; Blocked=randperm(N);
% bond
```

```
[p,q,r]=find(triu(nghc)); b=[p,q]; bn=size(b,1); bcm=sparse(bn,bn);
for i=1:bn,
  bcm(i,b(i,1))=1; bcm(i,b(i,2))=1;
end
nghb=sparse(bn,bn);
for i=1:cn,
  Tmp=find(bcm(:,i));
  if(~isempty(Tmp))
    TmpN=size(Tmp,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        nghb(Tmp(j),Tmp(k))=1; nghb(Tmp(k),Tmp(j))=1;
      end
    end
  end
end
N=bn; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:bn,
  if((cs(b(i,1),1)<=LBc) | (cs(b(i,2),1)<=LBc))
    LMat(1,i)=1;
  end
  if((cs(b(i,1),1)>=UBc) | (cs(b(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=nghb; Blocked=randperm(N);
% vertice
nghv=sparse(vn,vn);
for i=1:cn,
  Tmp=[c{i,3},c{i,3}(1)];
  for j=1:c{i,1},
    nghv(Tmp(j),Tmp(j+1))=1; nghv(Tmp(j+1),Tmp(j))=1;
  end
end
LMat=sparse(1,vn); UMat=sparse(1,vn);
for i=1:vn,
  if(vp(i,1)<=LBv)
    LMat(1,i)=1;
  end
  if(vp(i,1)>=UBv)
    UMat(1,i)=1;
  end
end
N=vn; NeMat=nghv; Blocked=randperm(N);
% edge
[p,q,r]=find(triu(nghv)); e=[p,q]; en=size(e,1); evm=sparse(en,en);
for i=1:en,
  evm(i,e(i,1))=1; evm(i,e(i,2))=1;
end
nghe=sparse(en,en);
for i=1:vn,
  Tmp=find(evm(:,i));
  if(~isempty(Tmp))
    TmpN=size(Tmp,1);
    for j=1:(TmpN-1),
      for k=(j+1):TmpN,
        nghe(Tmp(j),Tmp(k))=1; nghe(Tmp(k),Tmp(j))=1;
      end
    end
  end
end
N=en; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:en,
  if((vp(e(i,1),1)<=LBc) | (vp(e(i,2),1)<=LBc))
    LMat(1,i)=1;
  end
  if((vp(e(i,1),1)>=UBc) | (vp(e(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=nghe; Blocked=randperm(N);
```

§ A.5 Continuum percolation of n-gons

```
% regpoly.m
clear all; St=sum(100*clock); rand('state',St); Size=10; N=40; n=11; Angle=2*pi/n;
X=Size*rand(N,2); TwoPi=2*pi; Rad=TwoPi*rand(N,1);
R=sqrt(1/(n*sin(Angle/2)*cos(Angle/2)));
for i=1:N,
  for j=0:(n-1),
    Tmp=Rad(i,1)+j*Angle; V{i}{1,(j+1)}=[(X(i,1)+R*cos(Tmp)),(X(i,2)+R*sin(Tmp))];
  end
end
Tmp=V{1}{1,1}+(V{1}{1,2}-V{1}{1,1})/2; dx=Tmp(1,1)-X(1,1); dy=Tmp(1,2)-X(1,2);
r=sqrt(dx*dx + dy*dy); clf; hold on;
for i=1:N,
  x=[]; y=[];
  for j=1:n,
    x=[x;V{i}{1,j}(1,1)]; y=[y;V{i}{1,j}(1,2)];
  end
  x=[x;V{i}{1,1}(1,1)]; y=[y;V{i}{1,1}(1,2)]; plot(x,y);
end
plot([0,Size,Size,0,0],[0,0,Size,Size,0]); axis equal; axis off;
for i=1:N,
  Tmp=[];
  for j=1:n,
    Tmp=[Tmp;[j,V{i}{1,j}(1,2)]]; Tmp=sortrows(Tmp,2);
  end
end
T=delaunay(X(:,1),X(:,2)); NT=size(T,1); D=sparse(N,N); Ov=sparse(N,N);
Ov1=sparse(N,1); Pair=[]; Limbo=[]; Oclock=sparse(NT,NT);
for i=1:NT,
  Tmp=[T(i,:),T(i,1)];
  for j=1:3,
    c1=Tmp(1,j); c2=Tmp(1,(j+1)); dx=X(c2,1)-X(c1,1); dy=X(c2,2)-X(c1,2);
    TmpA=sqrt(dx*dx + dy*dy); D(c1,c2)=TmpA; D(c2,c1)=TmpA;
    Pair=[Pair;[c1,c2;c2,c1]];
    if(TmpA<=(2*r))
      Ov(c1,c2)=1; Ov(c2,c1)=1; Ov1(c1,1)=1; Ov1(c2,1)=1;
    elseif(TmpA<=(2*R))
      Limbo=[Limbo;[c1,c2;c2,c1]];
    end
  end
  TmpB=atan(abs(dy/dx));
  if(dx>=0)
    if(dy>=0)  % Quadrant 1
      Oclock(c1,c2)=TmpB; Oclock(c2,c1)=pi+TmpB;
    else  % Quadrant 4
      Oclock(c1,c2)=TwoPi-TmpB; Oclock(c2,c1)=pi-TmpB;
    end
  else
    if(dy>=0)  % Quadrant 2
      Oclock(c1,c2)=pi-TmpB; Oclock(c2,c1)=TwoPi-TmpB;
    else  % Quadrant 3
      Oclock(c1,c2)=pi+TmpB; Oclock(c2,c1)=TmpB;
    end
  end
end
Tmp=Angle/2; Star=[];
for i=1:N,
  TmpA=[]; TmpB=Rad(i,1);
  for j=1:n,
    TmpA=[TmpA,mod((TmpB + (j-1)*Angle + Tmp),TwoPi)];
  end
  Star=[Star;TmpA];
end
Wobble=sparse(N,N); jWobble=sparse(N,N); TmpN=size(Limbo,1);
for i=1:TmpN,
  Min=10; TmpA=Limbo(i,1); TmpB=Limbo(i,2); jMin=j;
  for j=1:n,
    Tmp=Star(TmpA,j)-Oclock(TmpA,TmpB);
    if(abs(Tmp)<abs(Min))
      Min=Tmp; jMin=j;
    end
  end
  Wobble(TmpA,TmpB)=Min; jWobble(TmpA,TmpB)=jMin;
end
Tmp=Angle/2;
for i=1:2:TmpN,
  TmpA=Limbo(i,1); TmpB=Limbo(i,2);
  if(abs(Wobble(TmpB,TmpA)) >= abs(Wobble(TmpA,TmpB)))
    TmpA=Limbo((i+1),1); TmpB=Limbo((i+1),2);
  end
```

```
  J=jWobble(TmpA,TmpB); v{1}=V{TmpA}{1,J};
  if(J==n)
    v{2}=V{TmpA}{1,1};
  else
    v{2}=V{TmpA}{1,(J+1)};
  end
  J=jWobble(TmpB,TmpA); v{3}=V{TmpB}{1,J};
  if(J==n)
    v{4}=V{TmpB}{1,1};
  else
    v{4}=V{TmpB}{1,(J+1)};
  end
  Max=0;
  if(Wobble(TmpA,TmpB)>=0)
    vMin=v{1};
  else
    vMin=v{2};
  end
  d1=R*cos(Tmp-abs(Wobble(TmpA,TmpB))); TmpD=(X(TmpA,1)-X(TmpB,1));
  a=(X(TmpA,2)-X(TmpB,2))/TmpD; b=(X(TmpA,1)*X(TmpB,2)-X(TmpB,1)*X(TmpA,2))/TmpD;
  a1=a; b1=vMin(1,2)-a1*vMin(1,1); x3=v{3}(1,1); y3=v{3}(1,2); x4=v{4}(1,1);
  y4=v{4}(1,2); TmpD=x3-x4; p=(y3-y4)/TmpD; q=(x3*y4-x4*y3)/TmpD; TmpD=a1-p;
  x=(q-b1)/TmpD; y=(a1*q-b1*p)/TmpD; dx=x-X(TmpB,1); dy=y-X(TmpB,2);
  r2=sqrt(dx*dx + dy*dy); d2=r2*cos(Tmp-abs(Wobble(TmpB,TmpA))); d=D(TmpA,TmpB);
  if((d1+d2)>=d)
    Ov(TmpA,TmpB)=1; Ov(TmpB,TmpA)=1; Ov1(TmpA,1)=1; Ov1(TmpB,1)=1;
  end
end
Clus=Ov;
for i=1:N,
  Clus(i,i)=1;
end
NClus=size(Clus,1); ClusA=Clus(1,:); NClusA=1;
for i=2:NClus,
  Joined=0;
  for j=1:NClusA,
    TmpC=Clus(i,:) | ClusA(j,:);
    if(sum(Clus(i,:) & ClusA(j,:)))
      ClusA(j,:)=TmpC; ClusB=ClusA; ClusA=TmpC; NClusB=NClusA;
      NClusA=1; Joined=1; break;
    end
  end
  if(~Joined)
    ClusA=[ClusA;Clus(i,:)]; NClusA=NClusA+1;
  else
    for j=1:NClusB,
      if(sum(ClusA(1,:) & ClusB(j,:)))
        ClusA(1,:)=ClusA(1,:) | ClusB(j,:);
      else
        ClusA=[ClusA;ClusB(j,:)]; NClusA=NClusA+1;
      end
    end
  end
end
Left=sparse(1,N); Right=sparse(1,N); Margin=0.1*Size;
for i=1:N,
  if(X(i,1)<=Margin)
    Left(1,i)=1;
  elseif(X(i,1)>=(Size-Margin))
    Right(1,i)=1;
  end
end
Plated=0;
for i=1:NClusA,
  if(sum(Left & ClusA(i,:)) & sum(Right & ClusA(i,:)))
    Plated=1; break;
  end
end
```

§ A.6 Tilings

```
% tiling.m;  NB. must run one of the data below first.
qn=size(q,1); in2n=size(in2,1); in3n=size(in3,1); p=[];
p{1,1}=[o;o]'; p{1,2}=sz; r=[]; s=[];
for i=1:sz,
  r(1,i)=dx(m(i)); s(1,i)=dy(n(i));
end
p{1,3}=r'; p{1,4}=s'; iin=size(ii,1); Tmp=ones(iin,1);
TmpA=sparse(Tmp,ii(:,1),Tmp,1,sz); Tmp=[]; TmpB=[]; TmpC=[]; count=0;
for i=1:sz,
  if(~TmpA(i))
    count=count+1; Tmp=[Tmp;i,count]; TmpB=[TmpB;r(i)]; TmpC=[TmpC;s(i)];
  end
end
p{2,1}=Tmp; p{2,2}=count; p{2,3}=TmpB; p{2,4}=TmpC; iiin=size(iii,1);
Tmp=ones(iiin,1); TmpA=sparse(Tmp,iii(:,1),Tmp,1,sz);
Tmp=[]; TmpB=[]; TmpC=[]; count=0;
for i=1:sz,
  if(~TmpA(i))
    count=count+1; Tmp=[Tmp;i,count]; TmpB=[TmpB;r(i)]; TmpC=[TmpC;s(i)];
  end
end
p{3,1}=Tmp; p{3,2}=count; p{3,3}=TmpB; p{3,4}=TmpC; iv=[ii,2*ones(iin,1)];
for i=1:iiin,
  Tmp=0;
  for j=1:iin,
    if(iii(i,1)==ii(j,1))
      Tmp=1;
    end
  end
  if(~Tmp)
    iv=[iv;iii(i,:),3];
  end
end
ivn=size(iv,1); Tmp=ones(size(ivn,1),1);
TmpA=sparse(Tmp,iv(:,1),Tmp,1,sz); Tmp=[]; TmpB=[]; TmpC=[]; count=0;
for i=1:sz,
  if(~TmpA(i))
    count=count+1; Tmp=[Tmp;i,count]; TmpB=[TmpB;r(i)]; TmpC=[TmpC;s(i)];
  end
end
p{4,1}=Tmp; p{4,2}=count; p{4,3}=TmpB; p{4,4}=TmpC; map=[]; Tmp=ones(sz,1);
map{1,1}=sparse(p{1,1}(:,1),Tmp,p{1,1}(:,2),sz,1); count=p{1,2}; e=q;
v=[p{1,3},p{1,4}]; Tmp=ones(p{2,2},1);
for i=2:nx,
  map{i,1}=sparse(p{2,1}(:,1),Tmp,count*Tmp+p{2,1}(:,2),sz,1);
  for k=1:iin,
    map{i,1}(ii(k,1),1)=map{(i-1),1}(ii(k,2),1);
  end
  count=count+p{2,2}; % +iin;
  for k=1:qn,
    e=[e;map{i,1}(q(k,1)),map{i,1}(q(k,2))];
  end
  for k=1:in2n,
    e=[e;map{i,1}(in2(k,1)),map{(i-1),1}(in2(k,2))];
  end
  v=[v;(i-1)*dim1*Tmp+p{2,3},p{2,4}];
end
Tmp=ones(p{3,2},1);
for j=2:ny,
  map{1,j}=sparse(p{3,1}(:,1),Tmp,count*Tmp+p{3,1}(:,2),sz,1);
  for k=1:iiin,
    map{1,j}(iii(k,1),1)=map{1,(j-1)}(iii(k,2),1);
  end
  count=count+p{3,2}; % +iiin;
  for k=1:qn,
    e=[e;map{1,j}(q(k,1)),map{1,j}(q(k,2))];
  end
  for k=1:in3n,
    e=[e;map{1,j}(in3(k,1)),map{1,(j-1)}(in3(k,2))];
  end
  v=[v;p{3,3},(j-1)*dim2*Tmp+p{3,4}];
end
Tmp=ones(p{4,2},1);
for i=2:nx,
  for j=2:ny,
    map{i,j}=sparse(p{4,1}(:,1),Tmp,count*Tmp+p{4,1}(:,2),sz,1);
    for k=1:ivn,
      if(iv(k,3)==2)
        map{i,j}(iv(k,1),1)=map{(i-1),j}(iv(k,2),1);
```

```
80        else
81          map{i,j}(iv(k,1),1)=map{i,(j-1)}(iv(k,2),1);
82        end
83      end
84      count=count+p{4,2}; % +ivn;
85      for k=1:qn,
86        e=[e;map{i,j}(q(k,1)),map{i,j}(q(k,2))];
87      end
88      for k=1:in2n,
89        e=[e;map{i,j}(in2(k,1)),map{(i-1),j}(in2(k,2))];
90      end
91      for k=1:in3n,
92        e=[e;map{i,j}(in3(k,1)),map{i,(j-1)}(in3(k,2))];
93      end
94      v=[v;(i-1)*dim1*Tmp+p{4,3},(j-1)*dim2*Tmp+p{4,4}];
95    end
96  end
97  en=size(e,1); vn=size(v,1); cn=size(c,1);
98  for i=1:cn,
99    c{i,2}=size(c{i},2);
100 end
101 C=[]; count=0;
102 for i=1:nx,
103   for j=1:ny,
104     for k=1:cn,
105       count=count+1;
106       C{count,1}=[];
107       for m=1:c{k,2},
108         C{count,1}=[C{count},map{i,j}(c{k,1}(m))];
109       end
110     end
111   end
112 end
113 ciin=size(cii,1);
114 for i=1:ciin,
115   cii{i,3}=size(cii{i,1},2); cii{i,4}=size(cii{i,2},2);
116 end
117 ciiin=size(ciii,1);
118 for i=1:ciiin,
119   ciii{i,3}=size(ciii{i,1},2); ciii{i,4}=size(ciii{i,2},2);
120 end
121 for i=2:nx,
122   for j=1:ciin,
123     count=count+1; C{count,1}=[];
124     for k=1:cii{j,3},
125       C{count,1}=[C{count},map{i,1}(cii{j,1}(k))];
126     end
127     for k=1:cii{j,4},
128       C{count,1}=[C{count},map{(i-1),1}(cii{j,2}(k))];
129     end
130   end
131 end
132 for i=2:ny,
133   for j=1:ciiin,
134     count=count+1;
135     C{count,1}=[];
136     for k=1:ciii{j,3},
137       C{count,1}=[C{count},map{1,i}(ciii{j,1}(k))];
138     end
139     for k=1:ciii{j,4},
140       C{count,1}=[C{count},map{1,(i-1)}(ciii{j,2}(k))];
141     end
142   end
143 end
144 civn=size(civ,1);
145 for i=1:civn,
146   civ{i,5}=size(civ{i,1},2); civ{i,6}=size(civ{i,2},2);
147   civ{i,7}=size(civ{i,3},2); civ{i,8}=size(civ{i,4},2);
148 end
149 for i=2:nx,
150   for j=2:ny,
151     for m=1:ciin,
152       count=count+1; C{count,1}=[];
153       for k=1:cii{m,3},
154         C{count,1}=[C{count},map{i,j}(cii{m,1}(k))];
155       end
156       for k=1:cii{m,4},
157         C{count,1}=[C{count},map{(i-1),j}(cii{m,2}(k))];
158       end
159     end
160     for m=1:ciiin,
161       count=count+1;
```

```
      C{count,1}=[];
      for k=1:ciii{m,3},
        C{count,1}=[C{count},map{i,j}(ciii{m,1}(k))];
      end
      for k=1:ciii{m,4},
        C{count,1}=[C{count},map{i,(j-1)}(ciii{m,2}(k))];
      end
    end
    for m=1:civn,
      count=count+1;
      C{count,1}=[];
      for k=1:civ{m,5},
        C{count,1}=[C{count},map{i,j}(civ{m,1}(k))];
      end
      for k=1:civ{m,6},
        C{count,1}=[C{count},map{(i-1),j}(civ{m,2}(k))];
      end
      for k=1:civ{m,7},
        C{count,1}=[C{count},map{(i-1),(j-1)}(civ{m,3}(k))];
      end
      for k=1:civ{m,8},
        C{count,1}=[C{count},map{i,(j-1)}(civ{m,4}(k))];
      end
    end
  end
end
Cn=size(C,1);
for i=1:Cn,
  C{i,2}=size(C{i,1},2);
end
x=[];
for i=1:Cn,
  Tmp=[0,0];
  for j=1:C{i,2},
    Tmp=Tmp+v(C{i,1}(j),:);
  end
  x=[x;Tmp/C{i,2}];
end
xn=size(x,1);
% for Cells
B=[]; Bxx=sparse(Cn,Cn); NeCMat=sparse(Cn,Cn); CVMat=sparse(Cn,vn);
for i=1:Cn,
  for j=1:C{i,2},
    CVMat(i,C{i,1}(j))=1;
  end
end
count=0;
for i=1:vn,
  Tmp=find(CVMat(:,i)); TmpN=size(Tmp,1);
  for j=1:(TmpN-1),
    for k=(j+1):TmpN,
      if(~NeCMat(Tmp(j),Tmp(k)))
        count=count+1; NeCMat(Tmp(j),Tmp(k))=1; NeCMat(Tmp(k),Tmp(j))=1;
        B=[B;Tmp(j),Tmp(k)]; Bxx(Tmp(j),Tmp(k))=count; Bxx(Tmp(k),Tmp(j))=count;
      end
    end
  end
end
Bn=count; A=x; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
LB=min(x(:,1)); UB=max(x(:,1)); rng=UB-LB;
% LBc=.05*rng+LB;
LBc=.1*rng+LB; UBc=UB-LBc;
for i=1:N,
  if(A(i,1)<=LBc)
    LMat(1,i)=1;
  elseif(A(i,1)>=UBc)
    UMat(1,i)=1;
  end
end
NeMat=NeCMat; Blocked=randperm(Cn);
% for Bonds
NeBMat=sparse(Bn,Bn);
for i=1:Cn,
  [p,q,r]=find(Bxx(i,:)); nc=size(r,2);
  for j=1:(nc-1),
    for k=(j+1):nc,
      NeBMat(r(1,j),r(1,k))=1; NeBMat(r(1,k),r(1,j))=1;
    end
  end
end
A=B; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
```

```
  if((x(A(i,1),1)<=LBc) | (x(A(i,2),1)<=LBc))
    LMat(1,i)=1;
  elseif((x(A(i,1),1)>=UBc) | (x(A(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=NeBMat; Blocked=randperm(Bn);
% for cells
b=[]; bxx=sparse(Cn,Cn); NecMat=sparse(Cn,Cn); count=0;
for i=1:(Cn-1),
  for j=(i+1):Cn,
    if(sum(CVMat(i,:) & CVMat(j,:))==2)
      count=count+1; NecMat(i,j)=1; NecMat(j,i)=1; b=[b;i,j];
      bxx(i,j)=count; bxx(j,i)=count;
    end
  end
end
bn=count; A=x; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if(A(i,1)<=LBc)
    LMat(1,i)=1;
  elseif(A(i,1)>=UBc)
    UMat(1,i)=1;
  end
end
NeMat=NecMat; Blocked=randperm(Cn);
% for bonds
NebMat=sparse(bn,bn);
for i=1:Cn,
  [a,bb,c]=find(bxx(i,:)); nc=size(c,2);
  for j=1:(nc-1),
    for k=(j+1):nc,
      NebMat(c(1,j),c(1,k))=1; NebMat(c(1,k),c(1,j))=1;
    end
  end
end
A=b; N=size(A,1); LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((x(A(i,1),1)<=LBc) | (x(A(i,2),1)<=LBc))
    LMat(1,i)=1;
  elseif((x(A(i,1),1)>=UBc) | (x(A(i,2),1)>=UBc))
    UMat(1,i)=1;
  end
end
NeMat=NebMat; Blocked=randperm(bn);
% for vertices
NeVMat=sparse(vn,vn);
for i=1:en,
  NeVMat(e(i,1),e(i,2))=1; NeVMat(e(i,2),e(i,1))=1;
end
A=v; N=vn; LMat=sparse(1,N); UMat=sparse(1,N); LB=min(v(:,1));
UB=max(v(:,1)); rng=UB-LB; LBv=.05*rng+LB; UBv=UB-LBv;
for i=1:vn,
  if(v(i,1)<LBv)
    LMat(1,i)=1;
  end
  if(v(i,1)>UBv)
    UMat(1,i)=1;
  end
end
NeMat=NeVMat; Blocked=randperm(vn);
% for edges
EVMat=sparse(en,vn);
for i=1:en,
  EVMat(i,e(i,1))=1; EVMat(i,e(i,2))=1;
end
NeEMat=sparse(en,en);
for i=1:vn,
  Tmp=find(EVMat(:,i)); TmpN=size(Tmp,1);
  for j=1:(TmpN-1),
    for k=(j+1):TmpN,
      NeEMat(Tmp(j),Tmp(k))=1; NeEMat(Tmp(k),Tmp(j))=1;
    end
  end
end
A=e; N=en; LMat=sparse(1,N); UMat=sparse(1,N);
for i=1:N,
  if((v(A(i,1),1)<=LBv) | (v(A(i,2),1)<=LBv))
    LMat(1,i)=1;
  elseif((v(A(i,1),1)>=UBv) | (v(A(i,2),1)>=UBv))
    UMat(1,i)=1;
  end
```

```
end
NeMat=NeEMat; Blocked=randperm(N);
% At the end of the day.
figure(1); clf; hold on;
for i=1:en,
  plot([v(e(i,1),1),v(e(i,2),1)],[v(e(i,1),2),v(e(i,2),2)]);
end
figure(2); clf; hold on;
for i=1:bn,
  plot([x(b(i,1),1),x(b(i,2),1)],[x(b(i,1),2),x(b(i,2),2)]);
end
figure(3); clf; hold on;
for i=1:Bn,
  plot([x(B(i,1),1),x(B(i,2),1)],[x(B(i,1),2),x(B(i,2),2)]);
end
% Here come data to be run first, though listed last.
% (1)  3_3[3^3]7_1[3^7]
clear all; sz=20; nx=7; ny=4; dx=0.5*[0;1;2;4;6;7;8;10;12]; Tmp=sqrt(3)/2;
dy=Tmp*[0;1;2;4;5;6;7;8;10;11;12]; dim1=max(dx); dim2=max(dy);
q=[1,2; 1,4; 2,3; 3,5; 4,7; 5,6; 5,8; 7,8; 7,9; 8,10; 9,12; 10,11; 10,14; 12,13;...
 13,14;,13,15; 14,16; 15,19; 16,17; 16,18; 18,20];
m=[3,4,6,1,7,9,3,5,2,7,8,2,3,5,1,7,9,6,3,4];
n=[1,1,2,3,3,3,4,4,5,6,6,7,8,8,9,9,9,10,11,11];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
ii=[4,6;15,17]; iii=[1,19;2,20]; in2=[9,11;12,11]; in3=[3,18];
c={[1,2,3,5,8,7,4];[7,8,10,14,13,12,9];[13,14,16,18,20,19,15]};
cii={[4,7,9],[11,10,8,5];[9,12],[11];[12,13,15],[16,14,10,11]};
ciii={[2,3],[18]}; civ={[1,4],[3,5],[18,16,17],[]};
% (2)  3_3[3^3]9_3[3^9]_II
clear all; sz=19; nx=4; ny=5; Tmp=1/sqrt(3); dx=Tmp*[0,1,2,3,4,5,6,7,8,9,10,11,12];
Tmp=1/3; dy=Tmp*[0,2,4,6,7,8,9,11,13,15,16,17,18]; dim1=max(dx); dim2=max(dy);
q=[1,3;2,4;3,5;3,6;4,7;4,9;5,6;5,9;6,8;7,10;8,11;9,12;10,13;12,15;...
 12,16;13,17;14,19;15,16;15,18;16,19;17,18];
m=[10,4,10,4,9,11,2,12,7,1,13,7,1,13,6,8,3,5,10];
n=[1,2,3,4,5,5,6,6,7,8,8,9,10,10,11,11,12,12,13];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19];
ii=[10,11;13,14]; iii=[1,19]; in2=[7,8]; in3=[2,17;2,18];
c={[3,6,5];[4,9,12,15,18,17,13,10,7];[12,16,15];[11,14,19,16,12,9,5,6,8]};
cii={[7,10],[8];}; ciii={[2],[17,18];[1,3,5,9,4,2],[18,15,16]};
civ={[2,4,7],[8,6,3,1],[14],[]};
% (3)  4_4[3^4]8_4[3^8]
clear all; sz=8; nx=10; ny=10; Tmp=1/sqrt(2); dx=[0,Tmp,1+Tmp,1+2*Tmp];
dy=[0,Tmp,1+Tmp,1+2*Tmp]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,3;2,4;3,5;5,7;6,8]; m=[2,3,1,4,1,4,2,3]; n=[1,1,2,2,3,3,4,4];
o=[1,2,3,4,5,6,7,8]; ii=[3,4;5,6]; iii=[1,7;2,8]; in2=[]; in3=[];
c={[1,2,4,6,8,7,5,3]}; cii={}; ciii={}; civ={[1,3],[2],[6],[]};
% (4)  3_3[3^3]8_2[3^8]
clear all; sz=25; nx=3; ny=6; dx=[0,1,2,3,4,5,6,7,8,9,10,11,12];
dy=.5*[0,2,4,5,6,8,10,11,12]; dim1=max(dx); dim2=max(dy);
q=[1,4;2,6;3,5;4,7;4,8;5,9;5,10;6,12;6,13;7,8;7,11;8,12;9,10;9,13;10,14;11,17;...
 12,15;13,16;15,19;15,20;16,21;16,22;17,23;18,25;19,20;19,23;20,24;21,22;21,24;22,25];
m=[3,7,11,3,11,7,2,4,10,12,1,5,9,13,5,9,1,13,4,6,8,10,3,7,11];
n=[1,1,1,2,2,3,4,4,4,4,5,5,5,5,6,6,7,7,8,8,8,8,9,9,9];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
ii=[11,14;17,18]; iii=[1,23;2,24;3,25]; in2=[]; in3=[];
c={[4,8,7];[5,10,9];[7,8,12,15,19,23,17,11];[6,13,16,21,24,20,15,12];...
 [9,10,14,18,25,22,16,13];[15,20,19];[16,22,21]}; cii={};
ciii={[2,6,12,8,4,1],[19,20];[3,5,9,13,6,2],[21,22]};
civ={[1,4,7,11],[10,5,3],[18],[]};
% (5)  3_3[3^3]9_3[3^9]_III
clear all; sz=12; nx=6; ny=4; Tmp=sqrt(3)/2; dx=Tmp*[0,1,2,3,4];
dy=.5*[0,3,5,6,9,11,12]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,3;2,3;3,4;4,5;4,6;5,7;6,8;7,9;9,11;10,12];
m=[2,4,3,3,2,4,1,5,1,5,2,4]; n=[1,1,2,3,4,4,5,5,6,6,7,7];
o=[1,2,3,4,5,6,7,8,9,10,11,12]; ii=[7,8;9,10]; iii=[1,11;2,12]; in2=[5,6]; in3=[];
c={[1,2,3];[4,6,8,10,12,11,9,7,5]}; cii={[5,7],[6]}; ciii={};
civ={[1,3,4,5],[6,4,3,2],[10],[]};
% (6)  4_3[3^4]10_6[3^10]_I
clear all; sz=20; nx=8; ny=4; dx=[0,2,3,5,6]; dy=3*[0,1,2,3,4];
dim1=max(dx); dim2=max(dy);
q=[1,2;1,6;2,3;2,7;3,4;4,5;6,7;6,9;7,10;9,10;10,11;11,12;11,14;12,13;12,15;14,15;14,18;15,19];
m=[1,2,3,4,5,1,2,5,1,2,3,4,5,3,4,1,2,3,4,5];
n=[1,1,1,1,1,2,2,2,3,3,3,3,3,4,4,5,5,5,5,5];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
ii=[1,5;6,8;9,13;16,20]; iii=[1,16;2,17;3,18;4,19;5,20]; in2=[]; in3=[];
c={[1,2,7,6];[2,3,4,5,8,13,12,11,10,7];[6,7,10,9];[11,12,15,14];[14,15,19,18]};
cii={[9,10,11,14,18,17,16],[19,15,12]}; ciii={}; civ={};
% (7)  3_3[3^3]9_3[3^9]_I
clear all; sz=37; nx=5; ny=3; x=2*sqrt(3)/(2+sqrt(3)); i=1/2; j=sqrt(3)/2;
dx=x*[0, i, i+j, 2*i+j, 3*i+j, 3*i+2*j, 4*i+2*j, 5*i+2*j, 5*i+3*j,...
 6*i+3*j, 7*i+3*j, 7*i+4*j, 8*i+4*j]; y=1/2/sqrt(3);
dy=x*[0, 2*y, 2*y+1, 4*y+1, 5*y+1, 5*y+1.5, 6*y+1.5, 8*y+1.5, 8*y+2.5, 10*y+2.5,...
```

```
 11*y+2.5, 11*y+3, 12*y+3, 14*y+3, 14*y+4, 16*y+4, 17*y+4, 17*y+4.5, 18*y+4.5,...
 20*y+4.5, 20*y+5.5, 22*y+5.5, 23*y+5.5, 23*y+6, 24*y+6]; dim1=max(dx); dim2=max(dy);
q=[1,4;2,3;3,5;3,6;4,8;4,9;5,6;5,7;6,9;7,10;8,11;9,14;10,12;12,15;13,16;14,17; 14,19;...
 15,19;16,18;17,18;17,20;18,20;19,22;20,21;21,23;21,24;22,25;22,27;23,24; 23,26;...
 24,28;25,26;25,29;26,29;27,31;29,30;30,33;30,34;31,35;32,37;33,34;33,36;34,37;35,36];
m=[10,4,4,10,3,5,2,12,7,1,13,1,13,7,2,12,9,11,4,10,10,4,9,...
 11,6,8,1,13,7,7,1,13,6,8,3,5,10];
n=[1,2,3,4,5,5,6,6,7,8,8,9,9,10,11,11,12,12,13,14,15,16,17,...
 17,18,18,19,19,20,21,22,22,23,23,24,24,25];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,...
 24,25,26,27,28,29,30,31,32,33,34,35,36,37]; ii=[10,11;12,13;27,28;31,32];
iii=[1,37]; in2=[7,8;15,16]; in3=[2,35;2,36];
c={[3,6,5];[5,6,9,14,19,15,12,10,7];[4,8,11,13,16,18,17,14,9];...
 [14,17,20,21,23,26,25,22,19];[17,18,20];[21,24,23];[22,25,29,30,33,36,35,31,27];...
 [23,24,28,32,37,34,30,29,26];[25,26,29];[30,34,33]};
cii={[7,10],[8];[12,15],[16];[15,19,22,27],[24,21,20,18,16]};
ciii={[2],[35,36];[1,4,9,6,3,2],[36,33,34]}; civ={[2,3,5,7],[8,4,1],[32],[35]};
% (8)  4_2[3^4]10_4[3^10]
clear all; sz=42; nx=3; ny=6; i=sqrt(3)-1; j=1/2; k=sqrt(3);
dx=[0, i, k-j, k, k+j, k+i, 2*k-i, 2*k-j, 2*k, 2*k+j, 3*k-i, 3*k, 3*k+i,...
 4*k-j, 4*k, 4*k+j, 4*k+i, 5*k-i, 5*k-j, 5*k, 5*k+j, 6*k-i, 6*k];
i=(2/3)*(3-sqrt(3)); j=1/sqrt(3); k=1/2/sqrt(3);
dy=[0, i, 2-k, 2, 2+j, 3, 3+i, 5-k, 5, 5+j, 6]; dim1=max(dx); dim2=max(dy);
q=[1,11;2,5;3,6;5,7;5,8;6,9;6,10;7,12;7,15;8,12;8,17;9,13;9,18;10,13;...
 10,20;11,21;12,16;13,19;14,23;15,16;15,21;16,17;17,22;18,19;18,22;19,20;...
 20,23;21,24;22,31;23,25;24,26;24,27;25,28;25,29;26,30;26,33;27,30;27,35;...
 28,32;28,36;29,32;29,38;30,34;31,40;31,41;32,37;33,34;33,39;...
 34,35;35,40;36,37;36,41;37,38;38,42];
m=[1,9,15,23,9,15,8,10,14,16,1,9,15,23,6,9,11,13,15,18,4,12,20,4,...
 20,3,5,19,21,4,12,20,2,4,7,17,20,22,1,9,15,23];
n=[1,1,1,1,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5,5,6,6,6,7,...
   7,8,8,8,8,9,9,9,10,10,10,10,10,10,11,11,11,11];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,...
 25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42];
ii=[1,4;11,14;39,42]; iii=[1,39;2,40;3,41;4,42]; in2=[]; in3=[];
c={[5,8,12,7];[6,10,13,9];[7,12,16,15];[8,17,16,12];[9,13,19,18];[10,20,19,13];...
 [15,16,17,22,31,40,35,27,24,21];[18,19,20,23,25,28,36,41,31,22];[24,27,30,26];...
 [25,29,32,28];[26,30,34,33];[27,35,34,30];[28,32,37,36];[29,38,37,32]};
cii={[11,21,24,26,33,39],[38,29,25,23]};
ciii={[2,5,7,15,21,11,1],[33,34,35];[3,6,9,18,22,17,8,5,2],[31];...
 [4,14,23,20,10,6,3],[36,37,38]}; civ={};
% (9)  4_3[3^4]10_6[3^10]_II
clear all; sz=56; nx=5; ny=3; i=2/(2-1/sqrt(3)); n=sqrt(3);
j=.5*(1.5*i-sqrt(3)); k=i/2; p=n-n*(1-i/n)-j; q=i;
dx=[0,j,k,p,q,n,n+j,n+k,n+p,n+q,2*n,2*n+j,2*n+k,2*n+p,2*n+q,3*n,3*n+j,...
 3*n+k,3*n+p,3*n+q,4*n]; m=3; j=i*(1/2+1/n)-1; tmp=1-i/sqrt(3);
p=i; k=p-tmp; q=p+tmp; r=q+i/2; s=2*i;
dy=[0,j,k,p,q,r,s,m,m+j,m+k,m+p,m+q,m+r,m+s,2*m,2*m+j,2*m+k,2*m+p,2*m+q,2*m+r,...
 2*m+s,3*m,3*m+j,3*m+k,3*m+p,3*m+q,3*m+r,3*m+s,4*m]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,7;2,8;3,6;5,6;6,10;7,8;7,12;8,13;10,11;10,14;11,18;11,19;12,13;13,16;...
 14,17;14,19;15,18;16,17;16,21;17,22;18,20;19,20;20,23;21,22;21,25;22,26;23,27;...
 24,28;25,26;25,28;26,29;27,30;27,31;28,32;29,30;29,33;30,34;32,35;33,34;33,38;...
 34,39;35,36;35,37;36,41;36,44;37,44;38,39;38,41;39,42;40,43;41,45;42,43;42,46;...
 43,47;44,45;45,48;46,47;46,51;47,52;48,49;48,50;49,55;50,54;51,52;51,55;52,56];
m=[1,5,15,21,9,13,1,5,21,13,17,1,5,11,21,6,10,20,14,18,6,10,18,2,6,10,16,...
   5,11,15,19,3,11,15,3,7,1,11,15,21,10,16,20,4,8,16,20,8,12,6,16,20,1,5,15,21];
n=[1,1,1,1,2,3,4,4,4,5,6,7,7,7,7,8,8,8,9,10,11,11,12,13,14,14,14,...
 15,15,15,16,17,18,18,19,20,21,21,21,21,22,22,22,23,24,25,25,26,27,28,28,28,29,29,29,29];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,...
 28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56];
ii=[1,4;7,9;12,15;37,40;53,56]; iii=[1,53;2,54;3,55;4,56];
in2=[24,23;24,31;32,31]; in3=[5,49;5,50];
c={[1,2,8,7];[7,8,13,12];[10,11,19,14];[11,18,20,19];[14,19,20,23,27,30,29,26,22,17];...
 [16,17,22,21];[21,22,26,25];[25,26,29,33,38,41,36,35,32,28];[29,30,34,33];...
 [33,34,39,38];[35,36,44,37];[36,41,45,44];[38,39,42,46,51,55,49,48,45,41];...
 [42,43,47,46];[46,47,52,51]};
cii={[12,13,16,21,25,28,24],[23,20,18];[24],[31,27,23];[24,28,32],[31];...
 [32,35,37],[43,42,39,34,30,27,31];[37,44,45,48,50,54,53],[52,47,43]};
ciii={[5,6,10,14,17,16,13,8,2],[50];[5],[50,48,49];[5,6,3],[49];...
 [4,9,15,18,11,10,6,3],[51,52]};civ={};
% (10)  4_2[3^4]8_2[3^8]
clear all; sz=32; nx=4; ny=6; i=1/2; j=sqrt(3);
dx=[0,i,2*i,j,2*j,3*j-2*i,3*j-i,3*j,3*j+i,3*j+2*i,4*j,5*j,6*j-2*i,6*j-i,6*j];
i=(2*(sqrt(3)-1))/sqrt(3); j=(sqrt(3)/2)+i; k=(1/2/sqrt(3))+j; m=i+sqrt(3);
n=3; dy=[0,i,j,k,m,n,n+i,n+j,n+k,n+m,2*n]; dim1=max(dx); dim2=max(dy);
q=[1,5;2,10;3,11;5,7;6,8;7,9;7,14;8,12;8,15;9,13;10,17;10,18;11,18;11,19;13,14;...
 14,17;15,16;15,19;17,23;18,20;19,25;20,21;20,22;21,24;21,26;22,24;22,28;23,29;...
 23,30;24,27;25,31;25,32;26,27;26,30;27,28;28,31];
m=[1,5,11,15,1,15,2,14,1,5,11,15,1,3,13,15,4,8,12,8,7,9,4,8,12,6,8,10,1,5,11,15];
n=[1,1,1,1,2,2,3,3,4,4,4,4,5,5,5,5,6,6,6,7,8,8,9,9,9,10,10,10,11,11,11,11];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,...
 24,25,26,27,28,29,30,31,32];
```

```
ii=[1,4;5,6;9,12;13,16;29,32]; iii=[1,29;2,30;3,31;4,32]; in2=[]; in3=[];
c={[7,14,13,9];[8,12,16,15];[10,18,20,21,26,30,23,17];[11,19,25,31,28,22,20,18];...
 [20,22,24,21];[21,24,27,26];[22,28,27,24]}; cii={[5,7,9],[8];[13,14,17,23,29],[25,19,15]};
ciii={[2,10,17,14,7,5,1],[23];[3,11,18,10,2],[26,27,28]};
civ={[1],[6,8,15,19,11,3],[25],[]};
% (11)  4_3[3^4]8_3[3^8]_I
clear all; sz=22; nx=9; ny=3; m=sqrt(3); i=2/(2-1/m); n=m/2; tmp=(m-i)/2;
tmp1=i/4; j=n-tmp1; k=n-tmp; p=n+tmp; q=n+tmp1;
dx=[0,j,k,p,q,m,m+j,m+k,m+p,m+q,2*m]; m=2; n=1; tmp=.5*(1-i+sqrt(3)*i/2); tmp1=2*tmp;
j=m-tmp1; k=j+i/2; p=m+tmp; q=m+n-tmp; r=p+i/2; s=m+n+tmp1; u=2*m+n; t=u+n;
dy=[0,j,k,p,q,r,s,u,t,t+j,t+k,t+p,t+q,t+r,t+s,t+u,2*t]; dim1=max(dx); dim2=max(dy);
q=[1,3;3,5;4,6;5,7;5,8;6,7;7,9;8,9;9,10;10,11;10,12;11,13;13,16;14,15;15,17;...
 15,18;16,17;17,19;18,19;19,20;20,21;20,22];
m=[1,11,1,11,5,9,8,2,6,6,1,11,1,11,7,3,4,10,6,6,1,11];
n=[1,1,2,2,3,4,5,6,7,8,9,9,10,10,11,12,13,14,15,16,17,17];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22];
ii=[1,2;3,4;11,12;13,14;21,22]; iii=[1,21;2,22]; in2=[8,6;16,18]; in3=[];
c={[5,7,9,8];[10,12,14,15,17,16,13,11];[15,18,19,17]};
cii={[3,5,8],[6];[8,9,10,11],[10,9,7,6];[13,16],[18,15];[16,17,19,20,21],[20,19,18]};
ciii={[2,4,6,7,5,3,1],[20]}; civ={};
% (12)  4_3[3^4]8_3[3^8]_II
clear all; sz=41; nx=5; ny=3; dx=[0,1,2,3,4,5,6,7,8,9,10,11,12];
dy=2.5*[0,1,2,3,4,5,6,7,8]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,10;2,3;3,4;4,5;4,8;5,6;5,9;6,7;8,9;8,12;9,13;10,11;11,12;11,20;12,13;13,14;...
 14,15;14,17;15,16;15,18;17,18;17,22;18,23;19,20;19,25;20,21;21,22;21,30;22,23;23,24;...
 24,26;25,27;26,32;27,28;28,29;28,33;29,30;29,34;30,31;31,32;31,40;33,34;33,36;34,37];
m=[1,3,5,6,8,10,13,6,8,1,4,6,8,9,11,13,9,11,2,4,7,9,...
 11,12,2,12,2,3,5,7,10,12,3,5,1,3,5,6,8,10,13];
n=[1,1,1,1,1,1,1,2,2,3,3,3,3,3,3,3,4,4,5,5,5,5,5,5,6,6,7,7,7,7,7,7,8,8,9,9,9,9,9,9,9];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,...
 28,29,30,31,32,33,34,35,36,37,38,39,40,41]; ii=[1,7;10,16;35,41];
iii=[1,35;2,36;3,37;4,38;5,39;6,40;7,41]; in2=[19,24;25,26;27,32]; in3=[];
c={[1,2,3,4,8,12,11,10];[4,5,9,8];[5,6,7,16,15,14,13,9];[8,9,13,12];...
 [11,12,13,14,17,22,21,20];[14,15,18,17];[17,18,23,22];[19,20,21,30,29,28,27,25];...
 [21,22,23,24,26,32,31,30,21];[28,29,34,33];[33,34,37,36]};
cii={[10,11,20,19],[24,23,18,15,16];[19,25],[26,24];[25,27],[32,26]};
ciii={[3,4,5,6],[34,29,30,31]}; civ={[1],[6],[31,32],[27,28,33,36]};
% (13)  4_3[3^4]8_3[3^8]_III
clear all; sz=22; nx=10; ny=3; x=16*sqrt(3)/(4*(sqrt(3)+1)); i=x/4; j=2*sqrt(3);
dx=[0,i,2*i,j-2*i,j-i,j,j+i,j+2*i,2*j-2*i,2*j-i,2*j]; i=(sqrt(3)/4)*x; j=12;
dy=[0,x,x+i,x+2*i,j-2-x-2*i,j-2-x-i,j-2-x,j-2,j,j+x,j+x+i,j+x+2*i,2*j-2-x-2*i,...
 2*j-2-x-i,2*j-2-x,2*j-2,2*j]; dim1=max(dx); dim2=max(dy);
q=[1,3;3,7;4,5;5,6;6,7;7,8;8,9;9,10;10,11;10,12;11,13;13,15;14,17;15,16;16,17;17,18;...
 18,19;19,20;20,21;20,22]; m=[1,11,1,11,10,9,8,7,6,6,1,11,1,11,2,3,4,5,6,6,1,11];
n=[1,1,2,2,3,4,5,6,7,8,9,9,10,10,11,12,13,14,15,16,17,17];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22];
ii=[1,2;3,4;11,12;13,14;21,22]; iii=[1,21;2,22]; in2=[8,5;9,6;15,18;16,19]; in3=[];
c={[10,12,14,17,16,15,13,11]};
cii={[3,7,8],[5];[8,9],[6,5];[9,10,11],[10,9,8,7,6];[13,15],[17,18];[15,16],[18,19];,,,
 [16,17,18,19,20,21],[19,20]}; ciii={[2,4,5,6,7,3,1],[20]}; civ={}
% (14)  4_4[3^4]7_2[3^7]_II
clear all; sz=33; nx=6; ny=5; dx=[0,1,2,3,4,5,6,7,8,9,10,11,12];
dy=4*[0,1,2,3,4]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,8;2,3;3,4;4,5;4,10;5,6;5,11;6,7;8,9;9,10;9,16;10,11;11,12;12,13;12,18;...
 13,14;13,19;15,16;15,21;16,17;17,18;17,24;18,19;19,20;20,26;21,22;22,23;22,28;...
 23,24;23,29;24,25;25,26;25,32];
m=[1,3,5,6,8,10,13,1,4,6,8,9,11,13,2,4,7,9,11,12,2,3,5,7,10,12,1,3,5,6,8,10,13];
n=[1,1,1,1,1,1,1,2,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5,5];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,...
 27,28,29,30,31,32,33]; ii=[1,7;8,14;27,33]; iii=[1,27;2,28;3,29;4,30;5,31;6,32;7,33];
in2=[15,20;21,26]; in3=[];
c={[1,2,3,4,10,9,8];[4,5,11,10];[5,6,7,14,13,12,11];[9,10,11,12,18,17,16];[12,13,19,18];...
 [15,16,17,24,23,22,21];[17,18,19,20,26,25,24];[22,23,29,28];[23,24,25,32,31,30,29]};
cii={[8,9,16,15],[20,19,13];[15,21],[26,20];[21,22,28,27],[32,25,26]}; ciii={}; civ={};
% (15)  3_3[3^3]12_6[3^12]
clear all; sz=17; nx=8; ny=5; i=2+sqrt(3); dx=[0,1,2,i,i+1,i+2,2*i];
i=sqrt(3); j=3+2*i; dy=[0,i,i+2,2*i+2,j,j+i,j+i+2,j+2*i+2,2*j]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,4;2,4;4,5;5,6;5,7;6,7;7,9;8,10;9,10;9,11;10,11;11,12;12,13;12,14;13,14;13,16;14,17];
m=[1,3,7,2,2,1,3,7,4,6,5,5,4,6,1,3,7]; n=[1,1,1,2,3,4,4,4,5,5,6,7,8,8,9,9,9];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17]; ii=[1,3;6,8;15,17]; iii=[1,15;2,16;3,17];
in2=[]; in3=[]; c={[1,2,4];[5,7,6];[9,10,11];[12,14,13]};
cii={[6,7,9,11,12,13,16],[17,14,12,11,10]}; ciii={};
civ={[1,4,5,6],[10,9,7,5,4,2],[13,14],[]};
% (16)  4_4[3^4]7_2[3^7]_I
clear all; sz=35; nx=6; ny=6; dx=[0,1,2,3,4,5,6]; dy=[0,1,2,3,4,5,6];
dim1=max(dx); dim2=max(dy);
q=[1,2;1,7;2,3;3,4;3,12;4,5;5,8;7,10;8,9;8,14;10,11;10,16;11,12;11,17;...
 12,13;13,14;13,18;14,19;16,17;17,22;18,19;18,23;19,20;20,21;20,28;22,23;...
 22,25;23,26;24,25;24,30;25,31;26,27;26,33;27,28;27,34;28,29];
m=[1,2,3,4,5,7,1,5,7,1,2,3,4,5,7,1,2,4,5,6,7,2,4,1,2,4,5,6,7,1,2,3,4,5,7];
n=[1,1,1,1,1,1,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,5,5,6,6,6,6,6,6,7,7,7,7,7,7];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,...
```

```
 27,28,29,30,31,32,33,34,35]; ii=[7,9;16,21;24,29];
iii=[1,30;2,31;4,33;5,34]; in2=[]; in3=[];
c={[1,2,3,12,11,10,7];[3,4,5,8,14,13,12];[13,14,19,18];[11,12,13,18,23,22,17];...
 [10,11,17,16];[18,19,20,28,27,26,23];[24,25,31,30];[26,27,34,33]};
cii={[16,17,22,25,24],[28,20];[10],[21,20,19,14,8,9]};
ciii={[2,3,4],[25,22,23,26];}; civ={[1],[9,8,5],[27,28],[24]};
% (17)  5_4[3^5]7_4[3^7]_I
clear all; sz=75; nx=2; ny=3; i=sqrt(3)/2;
dx=i*[0,1,2,3,4,6,7,8,9,10,12,13,14,15,16,18,19,20,21,22,24];
dy=.5*[0,1,2,4,6,7,8,10,12,13,14,16,18,19,20,22,24];
dim1=max(dx); dim2=max(dy);
q=[1,7;1,10;2,10;2,11;3,11;3,12;4,12;4,13;5,13;5,14;6,8;6,15;7,9;7,25;8,14;9,18;...
 10,19;11,20;12,16;13,17;14,23;16,17;16,21;17,22;18,27;19,25;19,28;20,26;20,29;...
 21,29;21,30;22,30;22,31;23,31;23,32;24,32;25,27;26,28;26,41;27,35;28,36;29,37;...
 30,38;31,33;32,34;33,34;33,39;34,40;35,43;35,44;36,44;36,45;37,41;37,46;38,42;...
 38,47;39,47;39,48;40,48;40,49;41,45;42,46;42,60;43,50;44,51;45,55;46,56;47,57;...
 48,58;50,51;50,53;51,54;53,62;54,62;54,63;55,63;55,64;56,64;56,65;57,60;57,66;...
 58,61;58,67;59,67;60,65;61,66;62,70;63,68;64,69;65,73;66,74;67,75;68,69;68,71;69,72];
m=[3,6,10,13,16,20,2,19,1,5,8,11,15,18,21,11,15,1,5,8,11,15,18,21,4,7,3,6,10,13,...
 16,20,16,20,3,6,10,13,16,20,9,12,1,5,8,11,15,18,21,1,5,21,1,5,8,11,15,18,21,14,...
 17,3,6,10,13,16,20,6,10,3,6,10,13,16,20];
n=[1,1,1,1,1,1,2,2,3,3,3,3,3,3,3,4,4,5,5,5,5,5,5,5,6,6,7,7,7,7,7,7,8,8,9,9,9,9,9,...
 9,10,10,11,11,11,11,11,11,11,12,12,12,13,13,13,13,13,13,13,14,14,15,15,15,15,15,...
 15,16,16,17,17,17,17,17,17];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,...
 30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,...
 56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75];
ii=[9,15;18,24;43,49;50,52;53,59]; iii=[1,70;2,71;3,72;4,73;5,74;6,75];
in2=[]; in3=[8,61];
c={[1,10,19,25,7];[2,11,20,26,28,19,10];[3,12,16,21,29,20,11];[4,13,17,16,12];...
 [5,14,23,31,22,17,13];[6,15,24,32,23,14,8];[7,25,27,18,9];[16,17,22,30,21];...
 [19,28,36,44,35,27,25];[20,29,37,41,26];[21,30,38,42,46,37,29];[22,31,33,39,47,38,30];...
 [23,32,34,33,31];[26,41,45,36,28];[33,34,40,48,39];[35,44,51,50,43];...
 [36,45,55,63,54,51,44];[37,46,56,64,55,45,41];[38,47,57,60,42];[39,48,58,61,66,57,47];...
 [40,49,52,59,67,58,48];[42,60,65,56,46];[50,51,54,62,53];[55,64,69,68,63]};
cii={[18,27,35,43],[40,34,32]};
ciii={[2,10,1],[62,54,63,68];[2,11,3],[69,68];[4,12,3],[69,64,56,65];...
 [5,13,4],[65,60,57,66];[5,14,8],[61,66];[8,6],[67,58,61]};
civ={[1,7,9],[6],[67,59],[62]};
% (18)  3_1[4^3]5_1[4^5]_I
clear all; sz=27; nx=4; ny=8; dx=[0,1,2,3,4,5,6,7,8]; dy=[0,1,2,3,4];
dim1=max(dx); dim2=max(dy);
q=[1,2;1,8;2,3;2,8;3,4;3,12;4,5;4,9;5,6;5,16;6,7;6,10;8,11;9,13;9,14;9,15;11,12;11,18;...
 12,13;12,23;13,14;13,19;14,15;14,19;15,16;15,19;16,17;16,25;18,21;18,22;19,24;20,26];
m=[1,2,3,5,7,8,9,1,5,9,1,3,4,5,6,7,9,1,5,9,1,2,3,5,7,8,9];
n=[1,1,1,1,1,1,1,2,2,2,3,3,3,3,3,3,3,4,4,4,5,5,5,5,5,5,5];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27];
ii=[1,7;8,10;11,17;18,20;21,27]; iii=[1,21;2,22;3,23;4,24;5,25;6,26;7,27]; in2=[]; in3=[];
c={[1,2,8];[2,3,12,11,8];[3,4,9,13,12];[4,5,16,15,9];[5,6,10,17,16];[6,7,10];...
 [9,14,13];[9,15,14];[11,12,23,22,18];[12,13,19,24,23];[13,14,19];[14,15,19];...
 [15,16,25,24,19];[16,17,20,26,25];[18,22,21]}; cii={}; ciii={[6,7],[20]}; civ={};
% (19)  3_1[4^3]5_1[4^5]_II
clear all; sz=13; nx=8; ny=8; dx=[0,1,2,3,4]; dy=[0,1,2,3,4]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,5;2,3;2,4;4,6;4,7;4,8;5,6;5,11;6,7;6,10;7,8;7,10;8,9;8,10;10,12];
m=[1,3,5,3,1,2,3,4,5,3,1,3,5]; n=[1,1,1,2,3,3,3,3,3,4,5,5,5];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13]; ii=[1,3;5,9;11,13]; iii=[1,11;2,12;3,13]; in2=[]; in3=[];
c={[1,2,4,6,5];[2,3,9,8,4];[4,7,6];[4,8,7];[5,6,10,12,11];[6,7,10];[7,8,10];[8,9,13,12,10]};
cii={}; ciii={}; civ={};
% (20)  5_3[3^5]8_6[3^8]_II
clear all; sz=110; nx=3; ny=2; i=sqrt(3)/2;
dx=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16];
dy=0.5*[0,1,2,4,6,7,8,10,12,13,14,16,18,19,20,22,24,25,26,28,30,31,32,34,...
 36,37,38,40,42,43,44,46,48]; dim1=max(dx); dim2=max(dy);
q=[1,5;1,8;2,6;2,9;3,9;3,10;4,10;4,11;5,7;5,19;6,8;7,14;8,15;9,12;10,13;12,13;...
 12,16;13,17;14,21;15,19;15,22;16,22;16,23;17,23;17,24;18,20;19,21;20,24;21,27;...
 22,25;23,26;24,30;25,26;25,28;26,29;27,31;27,34;28,34;28,35;29,35;29,36;30,32;...
 30,37;31,33;32,36;32,46;33,40;34,38;35,39;36,43;38,39;38,41;39,42;40,47;41,47;...
 41,48;42,48;42,49;43,45;43,50;44,46;45,49;45,58;46,50;47,51;48,52;49,55;50,56;...
 51,52;51,53;52,54;53,59;53,60;54,60;54,61;55,57;55,62;56,58;56,63;57,61;57,73;...
 58,62;59,64;60,65;61,69;62,70;64,65;64,67;65,68;67,74;68,74;68,75;69,72;69,76;...
 70,73;70,77;71,77;72,75;72,85;73,76;74,78;75,81;76,82;77,79;78,80;79,83;80,86;...
 80,87;81,84;81,88;82,85;82,89;83,89;83,90;84,87;84,100;85,88;86,91;87,95;88,96;...
 89,92;91,94;92,93;92,97;94,101;95,99;95,102;96,100;96,103;97,103;97,104;98,104;...
 99,101;100,102;101,107;102,108;103,105;104,106;105,106;105,109;106,110];
m=[3,7,11,15,2,6,1,5,9,13,17,9,13,1,5,9,13,17,4,16,3,7,11,15,7,11,3,7,...
 11,15,2,14,1,5,9,13,17,5,9,1,5,9,13,17,12,16,3,7,11,15,3,7,3,7,...
 11,15,10,14,1,5,9,13,17,1,5,17,1,5,9,13,17,8,12,3,7,11,15,3,15,3,...
 7,11,15,6,10,1,5,9,13,17,1,13,17,1,5,9,13,17,4,8,3,7,11,15,...
 11,15,3,7,11,15];
n=[1,1,1,1,2,2,3,3,3,3,3,4,4,5,5,5,5,5,6,6,7,7,7,7,8,8,9,...
 9,9,9,10,10,11,11,11,11,11,12,12,13,13,13,13,13,14,14,15,15,15,15,16,16,17,...
 17,17,17,18,18,19,19,19,19,19,20,20,20,21,21,21,21,21,22,22,23,23,23,23,24,24,...
```

```
 25,25,25,25,26,26,27,27,27,27,27,28,28,28,29,29,29,29,29,30,30,31,31,31,...
 31,32,32,33,33,33,33];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,...
 28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,...
 53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,...
 78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,...
 102,103,104,105,106,107,108,109,110];
ii=[7,11;14,18;33,37;40,44;59,63;64,66;67,71;86,90;91,93;94,98];
iii=[1,107;2,108;3,109;4,110]; in2=[31,20;78,79]; in3=[6,99];
c={[1,8,15,19,5];[2,9,12,16,22,15,8,6];[3,10,13,12,9];[4,11,18,20,24,17,13,10];...
 [5,19,21,14,7];[12,13,17,23,16];[15,22,25,28,34,27,21,19];[16,23,26,25,22];...
 [17,24,30,32,36,29,26,23];[25,26,29,35,28];[27,34,38,41,47,40,33,31];...
 [28,35,39,38,34];[29,36,43,45,49,42,39,35];[30,37,44,46,32];[32,46,50,43,36];...
 [38,39,42,48,41];[41,48,52,51,47];[42,49,55,57,61,54,52,48];[43,50,56,58,45];...
 [45,58,62,55,49];[51,52,54,60,53];[53,60,65,64,59];[54,61,69,72,75,68,65,60];...
 [55,62,70,73,57];[57,73,76,69,61];[64,65,68,74,67];[68,75,81,84,87,80,78,74];...
 [69,76,82,85,72];[70,77,79,83,89,82,76,73];[72,85,88,81,75];...
 [80,87,95,99,101,94,91,86];[81,88,96,100,84];[82,89,92,97,103,96,88,85];...
 [83,90,93,92,89];[84,100,102,95,87];[92,93,98,104,97];[97,104,106,105,103]};
cii={[14,21,27,31],[20];[31,33],[30,24,20];[40,47,51,53,59],[56,50,46];...
 [67,74,78],[79,77];[64],[71,77,70,62,58,56,63];[78,80,86],[83,79]};
ciii={[1,8,6],[99,101];[2,6],[99,95,102];[2,9,3],[105,103,96,100,102];[3,10,4],[106,105]};
civ={[1,5,7],[4],[106,104,98],[101]};
% (21)  5_3[3^5]8_6[3^8]_III
clear all; sz=110; nx=4; ny=2; i=sqrt(3)/2;
dx=i*[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16];
dy=0.5*[0,1,2,4,6,7,8,10,12,13,14,16,18,19,20,22,24,25,26,28,30,31,32,34,...
 36,37,38,40,42,43,44,46,48]; dim1=max(dx); dim2=max(dy);
q=[1,7;1,8;2,5;2,9;3,6;3,10;4,10;4,11;5,8;6,9;6,20;7,12;8,13;9,17;10,18;12,13;...
 12,15;13,16;15,22;16,22;16,23;17,23;17,24;18,20;18,25;19,21;20,24;21,25;22,28;...
 23,26;24,27;25,31;26,27;26,29;27,30;28,32;28,35;29,33;29,36;30,36;30,37;31,37;31,38;...
 32,34;33,35;33,47;34,39;35,43;36,44;37,40;39,42;40,41;40,45;42,49;43,49;43,50;44,47;...
 44,51;45,48;45,52;46,52;47,50;48,51;48,60;49,53;50,54;51,57;52,58;53,54;53,55;54,56;...
 55,59;55,62;56,62;56,63;57,63;57,64;58,60;58,65;59,61;59,73;60,64;61,68;62,69;63,66;...
 64,67;66,67;66,70;67,71;68,75;69,73;69,76;70,74;70,77;71,77;71,78;72,78;73,75;74,76;...
 74,85;75,79;76,82;77,83;78,80;79,81;80,84;81,87;81,88;82,88;82,89;83,85;83,90;84,86;...
 84,91;85,89;86,90;86,100;87,94;88,92;89,93;90,97;92,93;92,95;93,96;94,101;95,99;...
 95,102;96,102;96,103;97,103;97,104;98,100;99,101;100,104;101,107;102,108;103,105;...
 104,106;105,106;105,109;106,110];
m=[3,7,11,15,6,10,1,5,9,13,17,1,5,17,1,5,9,13,17,12,16,3,7,11,15,7,11,3,7,11,15,2,...
 6,1,5,9,13,17,1,13,17,1,5,9,13,17,8,12,3,7,11,15,3,7,3,7,11,15,2,14,1,5,9,13,17,9,...
 13,1,5,9,13,17,4,8,3,7,11,15,3,15,3,7,11,15,10,14,1,5,9,13,17,5,9,1,5,9,13,17,4,...
 16,3,7,11,15,11,15,3,7,11,15];
n=[1,1,1,1,2,2,3,3,3,3,3,4,4,4,5,5,5,5,5,6,6,7,7,7,7,8,8,9,9,9,9,10,10,11,11,11,...
 11,11,12,12,12,13,13,13,13,13,13,14,14,15,15,15,15,15,16,16,17,17,17,17,17,18,18,19,19,19...
 ,19,19,20,20,21,21,21,21,21,22,22,23,23,23,23,24,24,25,25,25,25,26,26,27,27,27,...
 27,27,28,28,29,29,29,29,29,30,30,31,31,31,31,32,32,33,33,33,33];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,...
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,...
 59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,...
 87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110];
ii=[7,11;12,14;15,19;34,38;39,41;42,46;61,65;68,72;87,91;94,98];
iii=[1,107;2,108;3,109;4,110]; in2=[32,21;79,80]; in3=[5,99];
c={[1,8,13,12,7];[2,9,17,23,16,13,8,5];[3,10,18,20,6];[6,20,24,17,9];...
 [12,13,16,22,15];[16,23,26,29,33,35,28,22];[17,24,27,26,23];[18,25,31,37,30,27,24,20];...
 [26,27,30,36,29];[28,35,43,49,42,39,34,32];[29,36,44,47,33];[30,37,40,45,48,51,44,36];...
 [31,38,41,40,37];[33,47,50,43,35];[40,41,46,52,45];[43,50,54,53,49];...
 [44,51,57,63,56,54,50,47];[45,52,58,60,48];[48,60,64,57,51];[53,54,56,62,55];...
 [55,62,69,73,59];[56,63,66,70,74,76,69,62];[57,64,67,66,63];[58,65,72,78,71,67,64,60];...
 [59,73,75,68,61];[66,67,71,77,70];[69,76,82,88,81,79,75,73];[70,77,83,85,74];...
 [71,78,80,84,86,90,83,77];[74,85,89,82,76];[81,88,92,95,99,101,94,87];...
 [82,89,93,92,88];[83,90,97,103,96,93,89,85];[84,91,98,100,86];[86,100,104,97,90];...
 [92,93,96,102,95];[97,104,106,105,103]};
cii={[15,22,28,32],[21];[12],[19,21,25,18,10,4,11];[32,34],[31,25,21];...
 [42,49,53,55,59,61],[58,52];[68,75,79],[80,78];[79,81,87],[84,80]};
ciii={[1,8,5],[99,101];[5,2],[102,95,99];[2,9,6,3],[105,103,96,102];[3,10,4],[106,105]};
civ={[1,7],[4],[106,104,100,98],[101]};
% (22)  5_2[3^5]12_12[3^12]
clear all; sz=34; nx=5; ny=3; i=sqrt(3)/2; dx=i*[0,1,2,3,4,5,6,7,8];
dy=[0,.5,1,3,3.5,4,5,6,6.5,7,9,9.5,10,11,12]; dim1=max(dx); dim2=max(dy);
q=[1,3;1,6;2,4;2,6;3,5;3,11;4,7;4,12;5,8;6,9;8,13;9,11;9,12;10,14;11,13;12,14;...
 13,15;14,16;15,17;16,18;17,19;17,21;18,20;18,23;19,22;19,27;20,22;20,28;21,24;...
 22,25;24,27;25,29;25,30;26,28;27,29;28,30;29,31;30,32;31,32;31,33;32,34];
m=[3,7,2,8,1,5,9,1,5,9,4,6,3,7,3,7,3,7,4,6,1,5,9,1,5,9,2,8,3,7,3,7,3,7];
n=[1,1,2,2,3,3,3,4,4,4,5,5,6,6,7,7,8,8,9,9,10,10,10,11,11,11,12,12,13,13,14,14,15,15];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,...
 28,29,30,31,32,33,34]; ii=[5,7;8,10;21,23;24,26]; iii=[1,33;2,34]; in2=[15,16]; in3=[];
c={[1,6,9,11,3];[2,4,12,9,6];[3,11,13,8,5];[4,7,10,14,12];...
 [9,12,14,16,18,20,22,19,17,15,13,11];[17,19,27,24,21];[18,23,26,28,20];...
 [19,22,25,29,27];[20,28,30,25,22];[25,30,32,31,29]};
cii={[8,13,15],[16,14];[15,17,21],[18,16]}; ciii={[1,6,2],[32,31]};
civ={[1,3,5],[4,2],[32,30,28,26],[27,29,31]};
```

```
% (23)  5_4[3^5]7_4[3^7]_II
clear all; sz=21; nx=3; ny=5; i=sqrt(3)/2; dx=i*[0,1,2,3,4,6,7,8,9,10,12];
dy=[0,.5,1,3,3.5,4,6]; dim1=max(dx); dim2=max(dy);
q=[1,4;1,6;2,5;2,7;3,8;3,9;4,7;4,14;5,8;6,10;7,11;8,12;10,14;11,16;11,17;12,15;...
 12,17;13,18;14,16;15,18;16,19;17,20;18,21];
m=[3,6,10,4,7,1,5,8,11,1,5,8,11,2,9,3,6,10,3,6,10];
n=[1,1,1,2,2,3,3,3,3,4,4,4,4,5,5,6,6,6,7,7,7];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21];
ii=[6,9;10,13]; iii=[1,19;2,20;3,21]; in2=[]; in3=[5,15];
c={[1,4,14,10,6];[2,5,8,12,17,11,7];[3,9,13,18,15,12,8];[4,7,11,16,14]};
cii={}; ciii={[1,4,7,2],[17,11,16];[2,5],[15,12,17];[5,8,3],[18,15]};
civ={[1,6],[3],[18,13],[14,16]};
% (24)  5_3[3^5]8_6[3^8]_I
clear all; sz=30; nx=8; ny=6; i=sqrt(3)/2; dx=[0,1,2,3,4,5,6,7,8];
dy=[0,.5,1,3,3.5,4,6,6.5,7,9,9.5,10,12]; dim1=max(dx); dim2=max(dy);
q=[1,3;1,6;2,4;2,7;3,5;3,11;4,6;5,8;6,9;8,13;9,11;9,14;10,12;11,13;12,14;13,15;...
 14,16;15,17;15,20;16,18;16,21;17,19;18,20;18,26;19,22;20,23;21,24;22,27;23,25;...
 23,28;24,26;25,27;26,28;27,29;28,30];
m=[3,7,2,6,1,5,9,1,5,9,4,8,3,7,3,7,2,6,1,5,9,1,5,9,4,8,3,7,3,7];
n=[1,1,2,2,3,3,3,4,4,4,5,5,6,6,7,7,8,8,9,9,9,10,10,10,11,11,12,12,13,13];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30];
ii=[5,7;8,10;19,21;22,24]; iii=[1,29;2,30]; in2=[17,12]; in3=[4,25];
c={[1,6,9,11,3];[2,7,10,12,14,9,6,4];[3,11,13,8,5];[9,14,16,18,20,15,13,11];...
 [15,20,23,25,27,22,19,17];[16,21,24,26,18];[18,26,28,23,20]};
cii={[8,13,15,17],[12];[17,19],[16,14,12]};
ciii={[1,6,4],[25,27];[4,2],[28,23,25]}; civ={[1,3,5],[2],[28,26,24],[27]};
% (25)  4_2[3^4]12_6[3^12]
clear all; sz=21; nx=10; ny=6; i=sqrt(3)/2;
dx=[0,1-i,.5,1,1.5,1+i,2,1.5+i,2+i,2.5+i,2+2*i]; i=1/sqrt(3); j=sqrt(3)/6;
d=sqrt(3)+1.5; dy=[0,i,i+j,3*i,3*i+1,d,d+i,d+i+j,d+3*i,d+3*i+1,2*d];
dim1=max(dx); dim2=max(dy);
q=[1,2;1,6;2,3;2,5;3,7;5,6;5,7;6,8;7,8;8,9;9,10;9,11;11,12;11,14;12,13;13,14;...
 13,15;14,16;15,16;16,17;17,20;17,21];
m=[1,4,7,11,4,3,5,4,4,2,6,9,9,8,10,9,9,1,4,7,11];
n=[1,1,1,1,2,3,3,4,5,6,6,6,7,8,8,9,10,11,11,11,11];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21];
ii=[1,4;18,21]; iii=[1,18;2,19;3,20;4,21]; in2=[10,12;10,15]; in3=[];
c={[1,2,5,6];[2,3,7,5];[5,7,8,6];[11,12,13,14];[13,15,16,14]};
cii={[10],[15,13,12]}; ciii={};
civ={[1,6,8,9,10],[12,11,9,8,7,3],[17],[18];[3,2],[4],[17,16,15],[10,9,11,14,16,17,20]};
% (26)  4_2[3^4]18_12[3^18]
clear all; sz=36; nx=6; ny=3; i=sqrt(3); j=6/i; d=2*j+i;
dx=[0,i,j,d-j,d-i,d,d+i,d+j,2*d-j,2*d-i,2*d];
dy=[0,3,4,6,7,9,10,13,15,18,19,21,22,24,25,28,30]; dim1=max(dx); dim2=max(dy);
q=[1,3;2,4;3,5;3,8;4,6;4,9;5,7;7,8;8,11;9,10;9,13;11,12;11,15;12,13;12,14;13,16;...
 14,15;14,16;15,17;16,17;17,18;18,19;18,20;19,21;19,22;20,21;20,24;21,23;22,23;...
 22,26;23,24;24,27;25,26;25,29;26,31;27,28;27,32;29,31;30,32;31,33;32,34;33,35;34,36];
m=[1,11,2,10,1,11,1,3,9,11,4,6,8,6,5,7,6,6,5,7,6,4,6,8,1,3,9,11,1,11,2,10,1,11,1,11];
n=[1,1,2,2,3,3,4,4,4,4,5,5,5,6,7,7,8,9,10,10,11,12,12,12,...
 13,13,13,13,14,14,15,15,16,16,17,17];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,...
 25,26,27,28,29,30,31,32,33,34,35,36];
ii=[1,2;5,6;7,10;25,28;29,30;33,34;35,36]; iii=[1,35;2,36]; in2=[]; in3=[];
c={[3,8,7,5];[4,6,10,9];[11,12,14,15];[12,13,16,14];[14,16,17,15];[18,20,21,19];...
 [19,21,23,22];[20,24,23,21];[25,26,31,29];[27,28,30,32]};
cii={[1,3,5],[4];[7,8,11,15,17,18,19,22,26,25],[27,24,20,18,17,16,13,9];[29,31,33],[32]};
ciii={[1,3,8,11,12,13,9,4,2],[34,32,27,24,23,22,26,31,33]}; civ={};
% (27)  5_3[3^5]7_3[3^7]_I
clear all; sz=60; nx=4; ny=2; x=sqrt(3); dx=x*[0,1,2,3,4,5,6];
dy=[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18]; dim1=max(dx); dim2=max(dy);
q=[1,5;2,5;2,6;3,6;3,7;4,7;5,8;6,11;7,9;8,10;9,12;10,13;10,14;11,14;11,15;12,15;...
 12,16;13,17;14,18;15,22;17,18;17,20;18,21;20,24;21,24;21,25;22,25;22,26;23,26;...
 24,27;25,28;26,31;27,28;27,29;28,30;29,32;29,33;30,33;30,34;31,34;31,35;32,38;...
 33,36;34,37;36,37;36,39;37,40;38,42;39,42;39,43;40,43;40,44;41,44;42,47;43,45;...
 44,46;45,46;45,48;46,49;47,50;47,51;48,51;48,52;49,52;49,53;50,54;...
 51,58;52,55;54,57;55,56;55,59];
m=[1,3,5,7,2,4,6,2,6,2,4,6,1,3,5,7,1,3,7,1,3,5,7,2,4,6,2,4,2,4,6,1,3,5,7,3,...
 5,1,3,5,7,2,4,6,4,6,2,4,6,1,3,5,7,1,5,7,1,3,5,7];
n=[1,1,1,1,2,2,2,3,3,4,4,4,5,5,5,5,5,6,6,6,7,7,7,7,8,8,8,9,9,10,10,10,11,11,11,...
 11,12,12,13,13,13,13,14,14,14,15,15,16,16,16,17,17,17,17,18,18,18,19,19,19,19];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,...
 31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60];
ii=[1,4;13,16;17,19;20,23;32,35;38,41;50,53;54,56;57,60];
iii=[1,57;2,58;3,59;4,60]; in2=[8,9]; in3=[];
c={[2,6,11,14,10,8,5];[3,7,9,12,15,11,6];[10,14,18,17,13];[11,15,22,25,21,18,14];...
 [17,18,21,24,20];[21,25,28,27,24];[22,26,31,34,30,28,25];[27,28,30,33,29];...
 [29,33,36,39,42,38,32];[30,34,37,36,33];[31,35,41,44,40,37,34];[36,37,40,43,39];...
 [39,43,45,48,51,47,42];[40,44,46,45,43];[45,46,49,52,48];[49,53,56,55,52]};
cii={[1,5,8],[9,7];[8,10,13],[12,9];[17],[23,26,22,15,12,16];...
 [20,24,27,29,32],[31,26];[38,42,47,50],[49,46,44]};
ciii={[1,5,2],[51,47,50,54];[2,6,3],[55,52,48,51];[3,7,4],[56,55]}; civ={};
% (28)  5_3[3^5]7_3[3^7]_II
```

```
clear all; sz=32; nx=5; ny=3; nx=3; ny=2; x=sqrt(3)/2; dx=[0,2,3,4,6,7,8];
dy=0.5*[0,2,4,5,6,10,11,12,14,16,17,18,22,23,24]; dim1=max(dx); dim2=max(dy);
q=[1,4;2,8;4,5;4,7;5,10;7,11;8,10;8,12;9,12;10,11;11,13;12,14;13,16;13,17;14,15;
 14,17;15,18;15,20;16,21;17,19;19,20;19,22;20,24;21,25;22,25;22,26;23,24;24,26;
 25,27;26,28;27,29;27,30;28,31;28,32;29,31];
m=[1,4,7,1,2,7,1,4,7,3,2,5,2,5,6,1,4,7,4,5,1,4,7,6,2,5,2,5,3,1,4,7];
n=[1,1,1,2,2,2,3,3,3,4,5,5,6,6,7,8,8,8,9,9,10,10,10,11,12,12,13,13,14,15,15,15];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32];
ii=[1,3;4,6;7,9;16,18;21,23;30,32]; iii=[1,30;2,31;3,32]; in2=[]; in3=[5,29];
c={[4,5,10,11,7];[8,12,14,17,13,11,10];[13,17,19,22,25,21,16];[14,15,20,19,17];...
 [15,18,23,24,20];[19,20,24,26,22];[22,26,28,31,29,27,25]};
cii={[7,11,13,16],[15,14,12];[21,25,27,30],[28,26,24]};
ciii={[1,4,5],[29,27];[2,8,10,5],[29];[2,8,12,9,6,3],[28]}; civ={[1],[],[28,26,24,23],[25,27]};
% (29)  3_2[4^3]5_2[4^5]_I
clear all; sz=15; nx=6; ny=6; dx=[0,1,2,3,4]; dy=[0,1,2,3,4]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,6;2,3;2,5;3,4;3,5;5,7;5,8;6,7;6,10;7,8;7,10;8,9;8,13;10,12;11,14];
m=[1,3,4,5,3,1,2,3,5,1,5,1,3,4,5]; n=[1,1,1,1,2,3,3,3,3,4,4,5,5,5,5];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]; ii=[1,4;6,9;10,11;12,15]; iii=[1,12;2,13;3,14;4,15];
in2=[]; in3=[];
c={[1,2,5,7,6];[2,3,5];[3,4,9,8,5];[5,8,7];[6,7,10];[7,8,13,12,10];[8,9,11,14,13]};
cii={}; ciii={}; civ={[1],[3],[11],[]};
% (30)  3_2[4^3]5_2[4^5]_II
clear all; sz=41; nx=3; ny=3; dx=[0,1,2,3,4,5,6,7,8]; dy=[0,1,2,3,4,5,6,7,8];
dim1=max(dx); dim2=max(dy);
q=[1,6;2,3;2,8;3,4;3,13;4,5;4,9;5,9;6,7;6,15;7,8;7,11;8,11;8,12;9,10;9,14;11,12;...
 11,16;12,13;12,22;13,14;13,17;14,17;14,18;15,16;15,20;16,20;16,21;17,18;17,23;...
 18,19;18,27;20,21;21,22;21,29;22,23;22,25;23,25;23,26;24,28;25,26;25,30;26,27;26,35;...
 27,28;27,31;28,31;28,32;29,30;29,33;30,33;30,34;31,32;31,36;32,40;33,34;33,37;34,35;...
 35,36;35,38;36,38;36,39];
m=[1,5,6,8,9,1,3,4,8,9,3,4,6,7,1,2,6,7,9,1,2,4,5,9,4,5,7,8,2,3,7,8,2,3,5,6,1,5,6,8,9];
n=[1,1,1,1,1,2,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5,6,6,6,6,7,7,7,7,8,8,8,8,9,9,9,9,9];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,...
 29,30,31,32,33,34,35,36,37,38,39,40,41];
ii=[1,5;6,10;15,19;20,24;37,41]; iii=[1,37;2,38;3,39;4,40;5,41]; in2=[29,32]; in3=[7,34];
c={[2,3,13,12,8];[3,4,9,14,13];[4,5,9];[5,10,9];[6,7,11,16,15];[7,8,11];[8,12,11];...
 [9,10,19,18,14];[11,12,22,21,16];[12,13,17,23,22];[13,14,17];[14,18,17];[15,16,20];...
 [16,21,20];[17,18,27,26,23];[18,19,24,28,27];[21,22,25,30,29];[22,23,25];[23,26,25];...
 [25,26,35,34,30];[26,27,31,36,35];[27,28,31];[28,32,31];[29,30,33];[30,34,33];...
 [31,32,40,39,36];[35,36,38];[36,39,38]};
cii={[20,21,29],[32,28];[29,33,37],[40,32]}; ciii={[1,6,7],[34,33];[7,8,2],[35,34]}; civ={};
% (31)  3_3[4^3]5_3[4^5]
clear all; sz=9; nx=6; ny=7; dx=[0,2,3,4,6]; dy=[0,2.5,5]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,6;2,3;2,5;3,4;3,5;5,7;5,8]; m=[1,2,4,5,3,1,2,4,5]; n=[1,1,1,1,2,3,3,3,3];
o=[1,2,3,4,5,6,7,8,9]; ii=[1,4;6,9]; iii=[2,7;3,8;4,9]; in2=[]; in3=[];
c={[1,2,5,7,6];[2,3,5];[5,8,7]}; cii={[1,6],[8,5,3]}; ciii={}; civ={};
% (32)  3_1[4^3]6_2[4^6]_I
clear all; sz=21; nx=3; ny=3; dx=[0,2,3,5,7,8,10];
dy=[0,2,3,5,7,8,10]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,7;2,3;2,7;3,4;3,6;4,5;4,8;6,10;6,11;6,12;7,9;9,10;9,15;10,11;10,14;11,12;...
 11,14;12,13;12,14;14,19;15,17;15,18;16,20];
m=[1,3,4,5,7,4,1,7,1,2,4,6,7,4,1,7,1,3,4,5,7]; n=[1,1,1,1,1,2,3,3,4,4,4,4,4,6,5,5,7,7,7,7,7];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]; ii=[1,5;7,8;9,13;15,16;17,21];
iii=[1,17;2,18;3,19;4,20;5,21]; in2=[]; in3=[];
c={[1,2,7];[2,3,6,10,9,7];[3,4,8,13,12,6];[4,5,8];[6,11,10];[6,12,11];[9,10,14,19,18,15];...
 [10,11,14];[11,12,14];[12,13,16,20,19,14];[15,18,17]}; cii={[15,17],[20]}; ciii={}; civ={};
% (33)  3_1[4^3]6_2[4^6]_II
clear all; sz=12; nx=6; ny=7; dx=[0,1,3,5,6]; dy=[0,2,3,5]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,8;2,3;2,6;3,4;3,6;4,5;4,6;6,7;7,9;7,10;7,11]; m=[1,2,3,4,5,3,3,1,2,3,4,5];
n=[1,1,1,1,1,2,3,4,4,4,4,4]; o=[1,2,3,4,5,6,7,8,9,10,11,12]; ii=[1,5;8,12];
iii=[1,8;2,9;3,10;4,11;5,12]; in2=[]; in3=[];
c={[1,2,6,7,9,8];[2,3,6];[3,4,6];[7,10,9];[7,11,10]}; cii={[1,8],[11,7,6,4]}; ciii={}; civ={};
% (34)  3_2[4^3]6_4[4^6]
clear all; sz=7; nx=10; ny=7; dx=[0,1,2,4]; dy=[0,3,6]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,6;2,3;2,4;4,5;4,6]; m=[1,3,4,2,1,3,4]; n=[1,1,1,2,3,3,3]; o=[1,2,3,4,5,6,7];
ii=[5,7;1,3]; iii=[1,5;2,6;3,7]; in2=[]; in3=[]; c={[1,2,4];[4,5,6]};
cii={[1,4,5],[6,4,2]}; ciii={}; civ={};
% (35)  3_3[4^3]6_6[4^6]
clear all; sz=11; nx=8; ny=5; dx=[0,1,2,3,4]; y=sqrt(3); dy=y*[0,1,2,3,4]; dim1=max(dx);
dim2=max(dy); q=[1,2;2,3;2,4;3,4;4,6;4,7;5,6;5,8;6,7;6,8;8,9;8,10];
m=[1,3,5,4,1,3,5,2,1,3,5]; n=[1,1,1,2,3,3,3,4,5,5,5]; o=[1,2,3,4,5,6,7,8,9,10,11];
ii=[1,3;5,7;9,11]; iii=[1,9;2,10;3,11]; in2=[]; in3=[]; c={[2,3,4];[4,7,6];[5,6,8];[8,10,9]};
cii={[1,2,4,6,5],[4];[5,8,9],[10,8,6]}; ciii={}; civ={};
% (36)  3_1[4^3]8_4[4^8]
clear all; sz=12; nx=7; ny=7; dx=[0,2,3,5]; dy=[0,2,3,5]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,5;2,3;2,5;3,4;3,6;5,7;7,9;7,10;8,11]; m=[1,2,3,4,1,4,1,4,1,2,3,4];
n=[1,1,1,1,2,2,3,3,4,4,4,4]; o=[1,2,3,4,5,6,7,8,9,10,11,12]; ii=[1,4;5,6;7,8;9,12];
iii=[1,9;2,10;3,11;4,12]; in2=[]; in3=[]; c={[1,2,5];[2,3,6,8,11,10,7,5];[3,4,6];[7,10,9]};
cii={[7,9],[11]}; ciii={}; civ={};
% (37)  3_1[5^3]4_2[5^4]_I
clear all; sz=8; nx=16; ny=4; dx=[0,1,2]; y=sqrt(3); dy=[0,2,y+2,y+4,2*y+4]; dim1=max(dx);
dim2=max(dy); q=[1,2;1,3;3,4;3,5;4,5;5,6;6,7;6,8]; m=[1,3,1,3,2,2,1,3]; n=[1,1,2,2,3,4,5,5];
o=[1,2,3,4,5,6,7,8]; ii=[1,2;3,4;7,8]; iii=[1,7;2,8]; in2=[5,5;6,6]; in3=[];
```

```
c={[1,2,4,3];[3,4,5];[6,8,7]}; cii={[3,5,6,7],[6,5]}; ciii={}; civ={};
% (38)  3_1[5^3]4_2[5^4]
clear all; sz=25; nx=3; ny=3; dx=[0,1,2,3,4]; dy=[0,1,2,3,4]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,6;2,3;2,6;2,7;3,4;3,7;3,8;4,5;4,9;6,7;6,11;7,8;7,12;8,9;8,12;8,13;9,10;9,13;...
 9,14;11,12;11,16;12,13;12,17;13,14;13,18;14,15;14,18;14,19;15,19;16,17;16,21;17,18;...
 17,21;17,22;18,19;18,23;19,20;19,24;20,24];
m=[1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5];
n=[1,1,1,1,1,2,2,2,2,2,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5];
o=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25];
ii=[1,5;6,10;11,15;16,20;21,25]; iii=[1,21;2,22;3,23;4,24;5,25]; in2=[]; in3=[];
c={[1,2,6];[2,7,6];[2,3,7];[3,8,7];[3,4,9,8];[4,5,10,9];[6,7,12,11];[7,8,12];[8,13,12];...
 [8,9,13];[9,14,13];[9,10,15,14];[11,12,17,16];[12,13,18,17];[13,14,18];[14,19,18];...
 [14,15,19];[15,20,19];[16,17,21];[17,22,21];[17,18,23,22];[18,19,24,23];[19,20,24]};
cii={[16,21],[24]}; ciii={}; civ={};
% (39)  3_2[5^3]4_4[5^4]
clear all; sz=12; nx=4; ny=4; t=37*pi/180; y=cos(t/2)/tan(t/2);
dx=[0,1,2,y+1,y+2,y+3,2*y+2]; dy=[0,1,y,y+1,y+2,2*y+1,2*y+2]; dim1=max(dx); dim2=max(dy);
q=[1,2;1,5;2,4;2,5;3,4;4,6;4,7;5,6;5,8;6,7;6,8;6,9;7,9;8,10;8,11;9,11;9,12];
m=[1,3,7,5,2,4,6,2,5,1,3,7]; n=[1,1,1,2,3,4,4,5,6,7,7,7]; o=[1,2,3,4,5,6,7,8,9,10,11,12];
ii=[1,3;10,12]; iii=[1,10;2,11;3,12]; in2=[5,7;8,7]; in3=[4,9];
c={[1,2,5];[2,4,6,5];[4,7,6];[5,6,8];[6,7,9];[6,9,11,8]};
cii={[1,5],[7,4];[5,8],[7];[8,10],[9,7];[8,11],[12]}; ciii={[2,4],[9];[4,3],[9]}; civ={};
```

§ A.7 Covering lattices

```
% cover contour
clear all; St=sum(100*clock); rand('state',St); CaN=100; X=rand(CaN,2);
[Va,Ca]=voronoin(X); VaN=size(Va,1); Vin=sparse(VaN,1);
for i=1:VaN,
  if((Va(i,1)<=1) & (Va(i,1)>=0) & (Va(i,2)<=1) & (Va(i,2)>=0))
    Vin(i,1)=1;
  end
end
C0=[]; Xn=[]; count=0;
for i=1:CaN,
  TmpN=size(Ca{i},2); Tmp=1;
  for j=1:TmpN,
    if(~Vin(Ca{i}(1,j),1))
      Tmp=0; break;
    end
  end
  if(Tmp)
    count=count+1; C0{count,1}=TmpN; C0{count,2}=Ca{i}; Xn=[Xn;i];
  end
end
CN=size(C0,1); C1=[];
for i=1:CN,
  Tmp=[C0{i,2},C0{i,2}(1,1)];
  for j=1:C0{i,1},
    C1{i,j}(1,1)=(Va(Tmp(1,j),1)+Va(Tmp(1,(j+1)),1))/2;
    C1{i,j}(1,2)=(Va(Tmp(1,j),2)+Va(Tmp(1,(j+1)),2))/2;
  end
end
C{1}=C1; figure(1); clf; hold on;
for i=1:CN,
  x=[]; y=[];
  for j=1:C0{i,1},
    x=[x,C{1}{i,j}(1,1)]; y=[y,C{1}{i,j}(1,2)];
  end
  x=[x,x(1,1)]; y=[y,y(1,1)]; plot(x,y);
end
axis equal;
n=8;
for k=2:n,
  Cn=[];
  for i=1:CN,
    TmpX=[]; TmpY=[];
    for j=1:C0{i,1},
      TmpX=[TmpX,C{k-1}{i,j}(1,1)]; TmpY=[TmpY,C{k-1}{i,j}(1,2)];
    end
    TmpX=[TmpX,TmpX(1,1)]; TmpY=[TmpY,TmpY(1,1)];
    for j=1:C0{i,1},
      Cn{i,j}(1,1)=(TmpX(1,j)+TmpX(1,(j+1)))/2;
      Cn{i,j}(1,2)=(TmpY(1,j)+TmpY(1,(j+1)))/2;
    end
  end
  C{k}=Cn;
end
figure(2); clf; hold on;
for i=1:CN,
```

```
  x=[]; y=[];
  for j=1:CO{i,1},
    x=[x,C{n}{i,j}(1,1)]; y=[y,C{n}{i,j}(1,2)];
  end
  x=[x,x(1,1)]; y=[y,y(1,1)]; plot(x,y);
end
axis equal; DEVV=sparse(VaN,VaN);
DVO=[];
for i=1:CN,
  TmpN=CO{i,1}; Tmp=[CO{i,2},CO{i,2}(1,1)];
  for j=1:TmpN,
    V1=Tmp(1,j); V2=Tmp(1,(j+1));
    if(~DEVV(V1,V2))
      dx=Va(V2,1)-Va(V1,1); dy=Va(V2,2)-Va(V1,2);
      TmpA=sqrt(dx*dx + dy*dy); DEVV(V1,V2)=TmpA; DEVV(V2,V1)=TmpA;
    end
    DVO{i,1}{1,1}(1,j)=DEVV(V1,V2); dx=Va(V1,1)-X(Xn(i,1),1);
    dy=Va(V1,2)-X(Xn(i,1),2); TmpA=sqrt(dx*dx + dy*dy); DVO{i,1}{2,1}(1,j)=TmpA;
  end
end
AO=[];
for i=1:CN,
  TmpN=CO{i,1}; Tmp=[DVO{i,1}{2,1},DVO{i,1}{2,1}(1,1)]; At=0;
  for j=1:TmpN,
    a=DVO{i,1}{1,1}(1,j); b=Tmp(1,j); c=Tmp(1,(j+1)); s=(a+b+c)/2;
    Ai=sqrt(s*(s-a)*(s-b)*(s-c)); At=At+Ai;
  end
  AO=[AO;At];
end
DV=[]; A=[];
for i=1:n,
  TmpA=[];
  for j=1:CN,
    TmpN=CO{j,1}; TmpB=[];
    for k=1:TmpN,
      TmpB=[TmpB;C{i}{j,k}];
    end
    TmpB=[TmpB;TmpB(1,:)];
    for k=1:TmpN,
      x1=TmpB(k,1); y1=TmpB(k,2); x2=TmpB((k+1),1); y2=TmpB((k+1),2);
      dx=x2-x1; dy=y2-y1; TmpA{j,1}(1,k)=sqrt(dx*dx + dy*dy);
      dx=x1-X(Xn(j,1),1); dy=y1-X(Xn(j,1),2); TmpA{j,2}(1,k)=sqrt(dx*dx+dy*dy);
    end
  end
  DV{i}=TmpA; TmpA=[];
  for j=1:CN,
    TmpN=CO{j,1}; TmpB=[DV{i}{j,2},DV{i}{j,2}(1,1)]; At=0;
    for k=1:TmpN,
      a=DV{i}{j,1}(1,k); b=TmpB(1,k); c=TmpB(1,(k+1)); s=(a+b+c)/2;
      Ai=sqrt(s*(s-a)*(s-b)*(s-c)); At=At+Ai;
    end
    TmpA=[TmpA;At];
  end
  A{i}=TmpA;
end
p=[sum(AO)];
for i=1:n,
  p=[p,sum(A{i})];
end
p=p*100/p(1); figure(3); clf; plot(0:n,p);
```

§ A.8 Covering contour

```
% gxy.m, cover contour (c) 2002, Kit Tiyapan@UMIST
clear all; St=sum(100*clock); rand('state',St); CaN=120;
X=rand(CaN,2); [Va,Ca]=voronoin(X); VaN=size(Va,1); Vin=sparse(VaN,1);
for i=1:VaN,
  if((Va(i,1)<=1) & (Va(i,1)>=0) & (Va(i,2)<=1) & (Va(i,2)>=0))
    Vin(i,1)=1;
  end
end
C0=[]; Xn=[]; count=0;
for i=1:CaN,
  TmpN=size(Ca{i},2); Tmp=1;
  for j=1:TmpN,
    if(~Vin(Ca{i}(1,j),1))
      Tmp=0; break;
    end
  end
  if(Tmp)
    count=count+1; C0{count,1}=TmpN; C0{count,2}=Ca{i}; Xn=[Xn;i];
  end
end
CN=size(C0,1); C1=[];
for i=1:CN,
  Tmp=[C0{i,2},C0{i,2}(1,1)];
  for j=1:C0{i,1},
    C1{i,j}(1,1)=(Va(Tmp(1,j),1)+Va(Tmp(1,(j+1)),1))/2;
    C1{i,j}(1,2)=(Va(Tmp(1,j),2)+Va(Tmp(1,(j+1)),2))/2;
  end
end
C{1}=C1;

n=8;
for k=2:n,
  Cn=[];
  for i=1:CN,
    TmpX=[]; TmpY=[];
    for j=1:C0{i,1},
      TmpX=[TmpX,C{k-1}{i,j}(1,1)]; TmpY=[TmpY,C{k-1}{i,j}(1,2)];
    end
    TmpX=[TmpX,TmpX(1,1)]; TmpY=[TmpY,TmpY(1,1)];
    for j=1:C0{i,1},
      Cn{i,j}(1,1)=(TmpX(1,j)+TmpX(1,(j+1)))/2;
      Cn{i,j}(1,2)=(TmpY(1,j)+TmpY(1,(j+1)))/2;
    end
  end
  C{k}=Cn;
end
```

§ A.9 Number of vertices

```
% numofvertices.m
clear all; dimmin=2; dimmax=9; batches=5; dvn=[]; cpu=[];
nmax=1000; rand('state',sum(100*clock));
for i=dimmin:dimmax,
  for j=1:batches,
    n=round(nmax/i); x=rand(n,i); t=cputime; [v,c]=voronoin(x);
    cpu(i,j)=(cputime-t)/n; lend=~floor(v); hend=~(ceil(v)-ones(size(v)));
    lhend=lend & hend; in=min(lhend,[],2); dvn(i,j)=sum(in)/n;
  end
end
dvn=[(1:dimmax)',dvn]; dvn=dvn(2:dimmax,:); figure(1); clf;
for i=1:batches,
    semilogy(dvn(:,1),dvn(:,(i+1)),'.','LineWidth',2); hold on;
end
edvn=[dvn(:,1),sum(dvn(:,2:(batches+1)),2)/batches]; tmp=edvn(:,2)./exp(edvn(:,1));
A=sum(tmp)/(dimmax-1); m=[dimmin,dimmax]; semilogy(m,A*exp(m));
cpu=[(1:dimmax)',cpu]; cpu=cpu(2:dimmax,:); figure(2); clf;
for i=1:batches,
    semilogy(cpu(:,1),cpu(:,(i+1)),'.','LineWidth',2); hold on;
end
ecpu=[cpu(:,1),sum(cpu(:,2:(batches+1)),2)/batches]; tmp=ecpu(:,2)./exp(ecpu(:,1));
B=sum(tmp)/(dimmax-1); m=[dimmin,dimmax]; semilogy(m,(B/35)*(exp(1)+2).^m);
```

§ A.10 Vertices per cell and cell ratio

```
% numveachcell.m
clear all; dimmin=2; dimmax=6; batches=5; nmax=3000;
rand('state',sum(100*clock));
for i=dimmin:dimmax,
  for j=1:batches,
    n=round(nmax*2/i); x=rand(n,i); [v,c]=voronoin(x);
    fleet{i,j,1}=v; fleet{i,j,2}=c; fleet{i,j,3}=n;
  end
end
for i=dimmin:dimmax,
  for j=1:batches,
    v=fleet{i,j,1}; c=fleet{i,j,2}; n=fleet{i,j,3}; lend=~floor(v);
    hend=~(ceil(v)-ones(size(v))); lhend=lend & hend; in=min(lhend,[],2);
    numvc=[]; vcin=[];
    for p=1:n,
      numvc=[numvc,size(c{p},2)]; flag=1;
      for q=1:numvc(p),
        if(~in(c{p}(q)))
          flag=0; break;
        end
      end
      if(flag)
        vcin=[vcin,numvc(p)];
      end
    end
    tmpn=size(vcin,2); rcin(i,j)=tmpn/n; vec(i,j)=sum(vcin)/tmpn;
  end
end
dum=rcin; str={'c_{in} / c_{all}'}; dum=dum(2:dimmax,:); tmp=[];
for i=dimmin:dimmax,
  tmp=[tmp;i*ones(batches,1),dum((i-1),:)'];
end
figure(1); semilogy(tmp(:,1),tmp(:,2),'.','LineWidth',2); hold on;
[p,s,mu]=polyfit(tmp(:,1),tmp(:,2),4); x=(dimmin:.02:dimmax)';
y=polyval(p,x,[],mu); semilogy(x,y); dum=vec; dum=dum(2:dimmax,:); tmp=[];
for i=dimmin:dimmax,
  tmp=[tmp;i*ones(batches,1),dum((i-1),dimmin:dimmax)'];
end
figure(2); semilogy(tmp(:,1),tmp(:,2),'.','LineWidth',2); hold on;
edum=[dum(:,1),sum(dum(:,2:(batches+1)),2)/batches];
tmp=edum(:,2)./exp(edum(:,1)); A=sum(tmp)/(dimmax-1); m=[dimmin,dimmax];
semilogy(m,(A/70)*(exp(1)+4).^m); xlabel('Dimension','FontSize',14);
ylabel(str,'FontSize',14);
```

§ A.11 TEX's macros

```
% thshead.tex
% thshead.tex
\def\Ordinate{\ifnum\day>30 1 \else\ifnum\day>20 \day-20 \else\day\fi\fi}
\def\date{{\number\day\/{\ifcase\Ordinate\or $^{st}$\or $^{nd}$\or $^{rd}$ \else $^{th}$\fi}}
  {\ifcase\month\or January\or February\or March\or April\or May\or June\or July\or August\or
    September\or October\or November\else December\fi}, {\number\year}}
\def\dat[#1:#2:#3]{\begingroup\tmp=#1 \ifnum\tmp>30 \tmp=1 \else\ifnum\tmp>20
 \advance\tmp-20 \fi\fi
 {#1\/{\ifcase\tmp\or $^{st}$\or $^{nd}$\or $^{rd}$ \else $^{th}$\fi}} #2 #3
 \endgroup}
\input manmac
% \tracingall
% \proofmodefalse
\ifproofmode\else\overfullrule=0pt\fi
\hsize=6in  \vsize=9.70820393249937in  \maxdepth=2pt \parindent=2pc
\pagewidth=\hsize \pageheight=\vsize \font\titlefont=cmbx10 at 15pt
\font\inchhigh=cminch at 20pt
\input epsf
\input rotate
\input ukhyphen
\def\home{/home/mjkpjkt2}
\def\dry#1{\ifcase#1\or
 \home/ar/thes1\or \home/ts\or \home/tiyapan/Extremum/DsgnExReport\or
 \home/xfg\or \home/ths\or \home/tiyapan/Perco/Papers\or \home/tiyapan/Extremum/Matlab\or
 \home/ar/rc1\or \home/ar/trans\or \home/ar/wn21\or
 \home/ar/wn23\or \home/ar/wn7\or \home/phy\or \home/rds\or \home/obj\or
 \home/voy\or \home/cum\or \home/flt\or \home/pcl\or \home/stp\fi}
\def\o[#1:#2]{$#1^{\hbox{\sevenrm #2}}$}
\def\Title{Ph.D. Thesis, UMIST. K N Tiyapan.}
\font\tenbm=cmmib10 \font\ninett=cmtt9 \font\sixit=cmti6 \font\twelvebf=cmbx12
\font\twelverm=cmr12 \def\[#1]{{\it #1}} \def\(#1){{\bf #1}}
\def\r[#1]{{\rm #1}} \def\s[#1]{{\sl #1}}
\def\beginsection#1\par{\vskip\z@ plus.3\vsize\penalty-250
 \vskip\z@ plus-.3\vsize\bigskip\vskip\parskip
 \message{#1}\leftline{\bf#1}\nobreak\smallskip\noindent}
\def\ifundefined#1{\expandafter\ifx\csname#1\endcsname\relax}
\def\ix[#1](#2){\expandafter\def\csname#1\endcsname{#2}}
\def\x[#1]{\ifundefined{#1}\ix[#1]($\aleph$)\fi \csname#1\endcsname}
\def\ipno[#1]{\ifundefined{#1p}\ix[#1p](1)\fi}
\def\pno[#1]{\immediate\write\ref{\string\ix[#1p](\number\pageno)}}
\input ref
\newwrite\ref \immediate\openout\ref=ref
\newdimen\dtmp \newdimen\dtmpi \newdimen\dtmpii \newcount\tmp
\def\Bf#1:{{\bf #1}} \def\ct{\hfil\par} \def\cut{\par\vfill\break}
\def\der[#1]{{\bf #1},} \def\hd#1:{\hfil{\twelvebf #1}\hfil}
\def\It#1:{{\it#1}} \def\pc{\hbox{Pc}}
\def\Sl#1:{{\sl#1}}
\def\w#1:{\dtmpi=\hsize \advance\dtmpi-.3em \dtmpii=\dtmpi \advance\dtmpii-.3em
  \parshape=2 0pt \dtmpi .3em \dtmpii {\tenpoint\bf #1.~}}
\def\vws[#1]#2:#3:{{\bf #1} ({\sl #2} in {\sl #3})}  % explain vowel sounds
\def\Lift#1#2{\raise#1em\hbox{#2}}
\def\beginindex{\begingroup
 \parindent=1em \maxdepth=\maxdimen
 \def\par{\endgraf \futurelet\next\inxentry}
 \obeylines \everypar={\hangindent 2\parindent}
 \exhyphenpenalty=10000 \raggedright}
\def\inxentry{\ifx\next\sub \let\next=\subentry
 \else\ifx\next\endindex \let\next=\vfill
 \else\let\next=\mainentry \fi\fi \next}
\def\endindex{\mark{}\break\endgroup}
\let\sub=\indent \newtoks\maintoks \newtoks\subtoks
\def\mainentry#1,{\mark{}\noindent
 \maintoks={#1}\mark{\the\maintoks}#1,}
\def\subentry\sub#1,{\mark{\the\maintoks}\indent
 \subtoks={#1}\mark{\the\maintoks\sub\the\subtoks}#1,}
\def\boxit#1{\vbox{\hrule\hbox{\vrule\kern3pt
  \vbox{\kern3pt#1\kern3pt}\kern3pt\vrule}\hrule}}
\def\bxt<#1>{\vbox{\hrule\hbox{\vrule\kern.7pt
  \vbox{\kern.7pt#1\kern.7pt}\kern.7pt\vrule}\hrule}}
\def\square#1#2{{\vcenter{\vbox{\hrule height.#2pt
  \hbox{\vrule width.#2pt height#1pt \kern#1pt
    \vrule width.#2pt}
  \hrule height.#2pt}}}}
\def\sqr{\mathchoice\square34\square34\square{2.1}3\square{1.5}3}
\def\Le{\;{\raise.3em\hbox{$<$}}{\kern-.8em\lower.3em\hbox{$=$}}\;}
\def\Ge{\;{\raise.3em\hbox{$>$}}{\kern-.8em\lower.3em\hbox{$=$}}\;}
\def\matrix#1{\null\,\vcenter{\normalbaselines\m@th
    \ialign{\hfil$##$\hfil&&\quad\hfil$##$\hfil\crcr
      \mathstrut\crcr\noalign{\kern-\baselineskip}
```

```
      #1\crcr\mathstrut\crcr\noalign{\kern-\baselineskip}}}\,}
\def\pmatrix#1{\left(\matrix{#1}\right)}
\def\Cgy{\mathop{\cal G}\nolimits}
\def\Cov[#1]{\hbox{${\rm C}\sp{#1}({\cal V})$}}
\def\Gl{\mathop{\raise.23em\hbox{$>$}\kern-.71em\lower.23em\hbox{$<$}}}
\def\Mod{\mathop{\rm mod}}
\def\Lift#1#2{\raise#1em\hbox{#2}}
\def\uncatcodespecials{\def\do##1{\catcode`##1=12 }\dospecials}
\newcount\lineno
\def\setupverbatim{\tt \lineno=0
  \def\par{\leavevmode\endgraf} \catcode`\`=\active
  \obeylines \uncatcodespecials \obeyspaces
  \everypar{\advance\lineno by1 \llap{\sevenrm\the\lineno\ \ }}}
{\catcode`\`=\active \gdef|{\char'174}\gdef`{\relax\lq}\obeyspaces\global\let =\ }
\def\listing#1{\par\smallskip\begingroup\setupverbatim\ninett\baselineskip.9em\input#1 \endgroup}
\def\tran#1{\begingroup\ninepoint\baselineskip.8em\parindent=1em\input#1\endgroup}
\def\picx#1:#2:{\epsfxsize=#1 \epsffile{#2.eps}}
\def\picy#1:#2:{\epsfysize=#1 \epsffile{#2.eps}}
\def\nin#1:{\begingroup\baselineskip1em\leftskip1em\parindent-1em % internet info
  \def\net##1:##2:##3:{{\tt ##1}{\it ##2}##3\par}\input #1\endgroup}
\def\bib#1:{\begingroup\baselineskip1em\leftskip1em\parindent-1em\input #1\endgroup}
\def\art#1:#2:#3:#4:{{\rm #1}~#2~{\it #3} #4\par}
\newcount\kpc
\def\kart#1[#2]#3:#4:#5:{\advance\kpc by 1 \ix[#2]({{\sevenrm KNT\hxdc(\romannumeral\the\kpc)}})
{\rm #1}~#3~{\it #4} #5 \x[#2].\par}
\def\bbib{\begingroup\baselineskip1em\leftskip1em\parindent-1em} \def\ebib{\endgroup}
\def\hxdc{{\count0=\count1 \divide\count1 by16
  \ifnum\count1>0 \hxdc\fi \count2=\count1 \multiply\count2 by-16
  \advance\count0 by\count2 \hexdigit}}
\def\hexdigit{\ifnum\count0<10 \number\count0
  \else\advance\count0 by-10 \advance\count0 by`A \char\count0 \fi}
% \pictri <file1>:<file2>:<file3>:<num1>:<num2>:<num3>:
\def\pictri#1:#2:#3:#4:#5:#6:{\smallskip
 \centerline{\vbox{\dtmp=\hsize \advance\dtmp-1em
  \dtmpi=\hsize \advance\dtmpi-2em \divide\dtmpi by3
  \halign to\dtmp{\hfil##\hfil&\hfil##\hfil&\hfil##\hfil\cr
   \epsfxsize=\dtmpi \epsffile{#1.eps} &\epsfxsize=\dtmpi \epsffile{#2.eps} &
   \epsfxsize=\dtmpi \epsffile{#3.eps} \cr
   (#4) &(#5) &(#6)\cr}}}}
% \picqua <file1>:<file2>:<file3>:<file4>:<num1>:<num2>:<num3>:<num4>:
\def\picqua#1:#2:#3:#4:#5:#6:#7:#8:{\smallskip
 \centerline{\vbox{\dtmp=\hsize \advance\dtmp-1em
  \dtmpi=\hsize \advance\dtmpi-2em \divide\dtmpi by4
  \halign to\dtmp{\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\hfil\cr
   \epsfxsize=\dtmpi \epsffile{#1.eps} &\ifx0#6 \else\epsfxsize=\dtmpi \epsffile{#2.eps}\fi
&
   \ifx0#7 \else\epsfxsize=\dtmpi \epsffile{#3.eps}\fi &\ifx0#8 \else\epsfxsize=\dtmpi
   \epsffile{#4.eps}\fi\cr
   (#5) &\ifx0#6 \else(#6)\fi &\ifx0#7 \else(#7)\fi &\ifx0#8 \else(#8)\fi\cr}}}}
% \picpent <file1>:<file2>:<file3>:<file4>:<file5>:<num>:
\def\picpent#1:#2:#3:#4:#5:#6:{\smallskip
 \centerline{\vbox{\dtmp=\hsize \advance\dtmp-1em
  \dtmpi=\hsize \advance\dtmpi-2em \divide\dtmpi by5
  \halign to\dtmp{\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\hfil\cr
   \epsfxsize=\dtmpi \epsffile{#1.eps} &\ifx0#2 \else\epsfxsize=\dtmpi \epsffile{#2.eps}\fi
&
   \ifx0#3 \else\epsfxsize=\dtmpi \epsffile{#3.eps}\fi &\ifx0#4 \else\epsfxsize=\dtmpi
   \epsffile{#4.eps}\fi &
   \ifx0#5 \else\epsfxsize=\dtmpi \epsffile{#5.eps}\fi\cr
   (a#6) &\ifx0#2 \else(b#6)\fi &\ifx0#3 \else(c#6)\fi &\ifx0#4 \else(d#6)\fi &\ifx0#5
   \else(e#6)\fi\cr}}}}
\def\picsept#1:#2:#3:#4:#5:#6:#7:#8:{\smallskip
 \centerline{\vbox{\dtmp=\hsize \advance\dtmp-1em
  \dtmpi=\hsize \advance\dtmpi-4em \divide\dtmpi by7
  \halign to\dtmp{\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\hfil&
    \hfil##\hfil\cr
   \epsfxsize=\dtmpi \epsffile{#8#1.eps} &
   \epsfxsize=\dtmpi \epsffile{#8#2.eps} &
   \epsfxsize=\dtmpi \epsffile{#8#3.eps} &
   \epsfxsize=\dtmpi \epsffile{#8#4.eps} &
   \epsfxsize=\dtmpi \epsffile{#8#5.eps} &
   \epsfxsize=\dtmpi \epsffile{#8#6.eps} &
   \epsfxsize=\dtmpi \epsffile{#8#7.eps} \cr}}}}
\def\picsepu#1:#2:#3:#4:{\smallskip
 \centerline{\vbox{\dtmp=\hsize \advance\dtmp-1em
  \dtmpi=\hsize \advance\dtmpi-4em \divide\dtmpi by7
  \halign to\dtmp{##\hfil&\hfil##\hfil&\hfil##\hfil&\hfil##\cr
   \epsfxsize=\dtmpi \epsffile{#4#1.eps} &
   \epsfxsize=\dtmpi \epsffile{#4#2.eps} &
   \epsfxsize=\dtmpi \epsffile{#4#3.eps} \cr}}}}
% \pclr <dim_x>:<filename>:<directory>:<index>:<caption>:
\def\pclr#1:#2:#3:#4:#5:{\begingroup\dtmp=\hsize \advance\dtmp-1em \dtmpi=\dtmp \advance\dtmpi-#1
```

```
  \advance\dtmpi-1em \smallskip \fig[#4]#5:
 \def\pcl{\centerline{\vbox{\hsize=\dtmp \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}\quad\hfil
  \vbox{\hsize=\dtmpi\noindent{\bf Figure \x[#4]} #5\smallskip}}}}}
 \centre{}
 \def\pcr{\centerline{\vbox{\hsize=\dtmp
  \hbox{\vbox{\hsize=\dtmpi\noindent{\bf Figure \x[#4]} #5\smallskip} \quad\hfil
  \epsfxsize=#1 \epsffile{#3#2.eps}}}}}
 \ipno[#4]
 \ifodd\csname#4p\endcsname \pcr \else \pcl \fi\ct\smallskip\endgroup\pno[#4]}
% \pplr <dim_x>:<filename>:<directory>:<index>:<caption>:<paragraph>:
\def\pplr#1:#2:#3:#4:#5:#6:{\begingroup\dtmp=\hsize \advance\dtmp-1em
  \dtmpi=\dtmp \advance\dtmpi-#1
 \advance\dtmpi-1em\smallskip \fig[#4]#5:
 \def\pcl{\centerline{\vbox{\hsize=\dtmp \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}\quad\hfil
  \vbox{\hsize=\dtmpi\parindent=0pt #6\hfil\smallskip\noindent{\bf Figure \x[#4]} #5.\smallskip}}}}}
 \def\pcr{\centerline{\vbox{\hsize=\dtmp
  \hbox{\vbox{\hsize=\dtmpi\noindent#6\hfil\smallskip\noindent\hfill{\bf Figure \x[#4]}
#5.
  \smallskip} \quad\hfil
  \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}}}}}}
 \ipno[#4]
 \ifodd\csname#4p\endcsname \pcr \else \pcl \fi\ct\smallskip\endgroup\pno[#4]}
\def\plr#1:#2:#3:#4:#5:#6:{\begingroup\dtmp=\hsize \advance\dtmp-1em \dtmpi=\dtmp
  \advance\dtmpi-#1 \advance\dtmpi-1em \smallskip \fig[#4]#5:
 \def\pcl{\centerline{\vbox{\hsize=\dtmp \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}\quad\hfil
  \vbox{\hsize=\dtmpi\noindent{#6\hfil\smallskip\bf Figure #4.} #5\smallskip}}}}}
 \def\pcr{\centerline{\vbox{\hsize=\dtmp
  \hbox{\vbox{\hsize=\dtmpi\noindent#6\hfil\smallskip{\bf Figure #4.} #5\smallskip} \quad\hfil
  \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}}}}}}
 \ipno[#2]
 \ifodd\csname#2p\endcsname \pcr \else \pcl \fi\ct\smallskip\endgroup\pno[#2]}
\def\plrx#1:#2:#3:#4:#5:#6:{\begingroup\dtmp=\hsize \advance\dtmp-1em \dtmpi=\dtmp
  \advance\dtmpi-#1 \advance\dtmpi-1em \smallskip
 \def\pcl{\centerline{\vbox{\hsize=\dtmp \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}\quad\hfil
  \vbox{\hsize=\dtmpi\noindent{#6\hfil\smallskip\bf #4.} #5\smallskip}}}}}
 \def\pcr{\centerline{\vbox{\hsize=\dtmp
  \hbox{\vbox{\hsize=\dtmpi\noindent#6\hfil\smallskip{\bf #4.} #5\smallskip} \quad\hfil
  \hbox{\epsfxsize=#1 \epsffile{#3#2.eps}}}}}}
 \ipno[#2]
 \ifodd\csname#2p\endcsname \pcr \else \pcl \fi\ct\smallskip\endgroup\pno[#2]}
\def\plrs#1:#2:#3:#4:#5:#6:{\begingroup\dtmp=\hsize \advance\dtmp-1em \dtmpi=\dtmp
 \advance\dtmpi-#1 \advance\dtmpi-1em \smallskip \fig[#4]#5:
 \def\pcl{\centerline{\vbox{\hsize=\dtmp \hbox{\epsfxsize=#1 \epsffile{#3#2.ps}\quad\hfil
  \vbox{\hsize=\dtmpi\noindent{#6\hfil\smallskip\bf #4.} #5\smallskip}}}}}
 \def\pcr{\centerline{\vbox{\hsize=\dtmp
  \hbox{\vbox{\hsize=\dtmpi\noindent#6\hfil\smallskip{\bf #4.} #5\smallskip} \quad\hfil
  \hbox{\epsfxsize=#1 \epsffile{#3#2.ps}}}}}}
 \ipno[#2]
 \ifodd\csname#2p\endcsname \pcr \else \pcl \fi\ct\smallskip\endgroup\pno[#2]}
% \twop:<xsize1>:<file1>:<label1>:<xsize2>:<file2>:<label2>:
\def\twop:#1:#2:#3:#4:#5:#6:{
  \centerline{\hfil\epsfxsize=#1 \epsffile{#2.eps}\hfil\epsfxsize=#4 \epsffile{#5.eps}\hfil}
  \centerline{\hfil #3\hfil\hfil\hfil #6\hfil}}
\def\tps:#1:#2:#3:#4:#5:#6:{
  \centerline{\hfil\epsfxsize=#1 \epsffile{#2.ps}\hfil\epsfxsize=#4 \epsffile{#5.ps}\hfil}
  \centerline{\hfil #3\hfil\hfil\hfil #6\hfil}}
\def\twap:#1:#2:#3:#4:{\begingroup\dtmp=\hsize \divide\dtmp by2 \advance\dtmp-1em
  \centerline{\hfil\epsfxsize=\dtmp \epsffile{#1.eps}\q\epsfxsize=\dtmp \epsffile{#3.eps}\hfil}
  \centerline{\hfil #2\hfil\hfil\hfil #4\hfil}\endgroup}
% \ppr<directory>:<fig1>:<fig1d>:<fig2>:<fig2d>{<capt1>}{<capt2>}
\def\ppr#1:#2:#3:#4:#5:#6#7{\begingroup\dtmp=\hsize \advance\dtmp-3em \dtmpi=\dtmp
  \divide\dtmpi by 4 \smallskip\parindent=0pt \dtmpii=0pt
 \setbox1=\vbox{\hsize=2\dtmpi #6} \setbox2=\vbox{\hsize=2\dtmpi #7}
 \ifdim\dtmpii<\ht1 \dtmpii=\ht1\fi \ifdim\dtmpii<\ht2 \dtmpii=\ht2\fi
 \centerline{\epsfxsize=\dtmpi \epsffile{#1#2.eps}~\epsfxsize=\dtmpi \epsffile{#1#3.eps}\hfil
  \epsfxsize=\dtmpi \epsffile{#1#4.eps}~\epsfxsize=\dtmpi \epsffile{#1#5.eps}}
 \centerline{\vbox to\dtmpii{\hsize=\dtmpi\unvbox1\vfil}\hfil
  \vbox to\dtmpii{\hsize=\dtmpi\unvbox2\vfil}}\smallskip
 \endgroup}
\def\prlr#1:#2:#3:#4{\begingroup\dtmp=\hsize \advance\dtmp-3em \dtmpi=\dtmp \divide\dtmpi
by 4
  \dtmp=\dtmpi \advance\dtmp1ex \dtmpii=\dtmpi \advance\dtmpii-1ex \smallskip\parindent=0pt
 \setbox1=\vbox{\hsize=2\dtmp \hfil\epsfxsize=\dtmpi \epsffile{#1#2.eps}
  ~\epsfxsize=\dtmpi \epsffile{#1#3.eps}\hfil}
 \setbox2=\vbox{\hsize=2\dtmpii #4\smallskip}
 \setbox3=\vbox{\hsize=2\dtmpii \hfill#4\smallskip}
 \def\pcl{\centerline{\copy1\hfil\copy3}} \def\pcr{\centerline{\copy2\hfil\copy1}}
 \ipno[#3]
 \ifodd\csname#3p\endcsname \pcr \else \pcl \fi\ct\endgroup\pno[#3]}
\def\ptlr#1:#2:#3:#4:#5{\begingroup\dtmp=\hsize \advance\dtmp-3em \dtmpi=\dtmp \divide\dtmpi
by 4
  \dtmp=\dtmpi \advance\dtmp1ex \dtmpii=\dtmpi \advance\dtmpii-1ex\smallskip\parindent=0pt
```

```
 238  \setbox1=\vbox{\hsize=3\dtmp \epsfxsize=\dtmpi \epsffile{#1#2.eps}
 239   ~\epsfxsize=\dtmpi \epsffile{#1#3.eps} ~\epsfxsize=\dtmpi \epsffile{#1#4.eps}}
 240  \setbox2=\vbox{\hsize=\dtmpii #5\smallskip}
 241  \setbox3=\vbox{\hsize=\dtmpii \hfill#5\smallskip}
 242  \def\pcl{\centerline{\copy1\hfil\copy3}} \def\pcr{\centerline{\copy2\hfil\copy1}}
 243  \ipno[#4]
 244  \ifodd\csname#4p\endcsname \pcr \else \pcl \fi\ct\endgroup\pno[#4]}
 245 \def\capt#1:#2:#3:{\nobreak\smallskip
 246  \centerline{\dtmp=\hsize \advance\dtmp-2em \vbox{\hsize \dtmp \noindent
 247  {\bf #1 #2} {\it #3}}}\smallskip}
 248 \def\balg{\begingroup\smallskip\begingroup\obeylines \sfcode`;=3000
 249  \def\b[##1]{{\bf ##1}} \def\i[##1]{{\it ##1}} \def\r[##1]{{\rm ##1}}}
 250 \def\ealg{\par\endgroup\endgroup\smallskip}
 251 \def\q{\quad} \def\qq{\qquad} \def\qqq{\qq\q} \def\qqqq{\qq\qq}
 252   \def\qqqqq{\qq\qq\q} \def\qqqqqq{\qq\qq\qq}
 253 \def\btoc{\begingroup\medskip\begingroup\parindent=0pt\obeylines \def\atpg{\dotfill}}
 254 \def\etoc{\endgroup\endgroup\medskip}
 255 \def\bdsc#1\edsc{\begingroup\smallskip\dtmp=\hsize \advance\dtmp-3em
 256   \centerline{\hfil\boxit{\vbox{\hsize=\dtmp #1}}\hfil}\par\endgroup\smallskip}
 257 \def\picl#1:#2:#3:#4:{\dtmp=\hsize \advance\dtmp-1em \dtmpi=\dtmp \advance\dtmpi-#1 \smallskip
 258  \centerline{\vbox{\halign to\dtmp{##\hfil&\hfil##\cr
 259  \epsfxsize=#1 \epsffile{#2.eps}&
 260  \vbox{\hsize=\dtmpi\noindent{\bf Figure \x[#3]} #4\smallskip}\cr}}}\smallski}
 261 \def\tbv#1:#2:#3:#4:#5:{\begingroup\smallskip
 262   \parindent0pt\centerline{\vbox{\dtmp=\hsize \advance\dtmp-2em
 263   \dtmpi=\dtmp \divide\dtmpi by5 \advance\dtmpi-.5em \def\h[##1]{\hskip.3em\sl##1}
 264   \setbox1=\vbox{\hsize=\dtmpi#1} \setbox2=\vbox{\hsize=\dtmpi#2}
 265   \setbox3=\vbox{\hsize=\dtmpi#3}
 266   \setbox4=\vbox{\hsize=\dtmpi#4} \setbox5=\vbox{\hsize=\dtmpi#5} \dtmpii=0pt
 267   \ifdim\dtmpii<\ht1 \dtmpii=\ht1\fi \ifdim\dtmpii<\ht2 \dtmpii=\ht2\fi
 268   \ifdim\dtmpii<\ht3 \dtmpii=\ht3\fi \ifdim\dtmpii<\ht4 \dtmpii=\ht4\fi
 269   \ifdim\dtmpii<\ht5 \dtmpii=\ht5\fi
 270   \halign to\dtmp{##\hfil&##\hfil&##\hfil&##\hfil&##\hfil\cr
 271   \vbox to\dtmpii{\hsize=\dtmpi\unvbox1\vfil} &\vbox to\dtmpii{\hsize=\dtmpi\unvbox2\vfil}
&
 272   \vbox to\dtmpii{\hsize=\dtmpi\unvbox3\vfil} &\vbox to\dtmpii{\hsize=\dtmpi\unvbox4\vfil}
&
 273   \vbox to\dtmpii{\hsize=\dtmpi\unvbox5\vfil}\cr}}}\endgroup}
 274 \def\tbiv#1:#2:#3:#4:{\begingroup\smallskip\parindent0pt
 275   \centerline{\vbox{\dtmp=\hsize \advance\dtmp-2em
 276   \dtmpi=\dtmp \divide\dtmpi by4 \advance\dtmpi-.5em \def\h[##1]{\hskip.3em\sl##1}
 277   \setbox1=\vbox{\hsize=\dtmpi#1} \setbox2=\vbox{\hsize=\dtmpi#2}
 278   \setbox3=\vbox{\hsize=\dtmpi#3}
 279   \setbox4=\vbox{\hsize=\dtmpi#4} \dtmpii=0pt
 280   \ifdim\dtmpii<\ht1 \dtmpii=\ht1\fi \ifdim\dtmpii<\ht2 \dtmpii=\ht2\fi
 281   \ifdim\dtmpii<\ht3 \dtmpii=\ht3\fi \ifdim\dtmpii<\ht4 \dtmpii=\ht4\fi
 282   \ifdim\dtmpii<\ht5 \dtmpii=\ht5\fi
 283   \halign to\dtmp{##\hfil&##\hfil&##\hfil&##\hfil\cr
 284   \vbox to\dtmpii{\hsize=\dtmpi\unvbox1\vfil} &\vbox to\dtmpii{\hsize=\dtmpi\unvbox2\vfil}
&
 285   \vbox to\dtmpii{\hsize=\dtmpi\unvbox3\vfil} &
 286   \vbox to\dtmpii{\hsize=\dtmpi\unvbox4\vfil}\cr}}}\endgroup}
 287 \def\tabhead{\def\erule{\vskip.2em\hrule\vskip.2em}
 288  \def\ptl{\quad} \def\ptr{\hfil\quad}}
 289 \def\trule{\noalign{\vskip.2em\hrule\vskip.2em}}
 290 \def\tbl#1\lbt{\begingroup\dtmp=\hsize \advance\dtmp-3em \smallskip
 291  \centerline{\vbox{\hsize=\dtmp \halign{##\hfil&&\quad##\hfil&\quad##\hfil\cr #1}}}\endgroup}
 292 \def\fgr[#1](#2)#3\rgf{\begingroup\dtmp=\hsize \divide\dtmp by#1 \advance\dtmp-1em
 293   \def\pic[##1]{\epsfxsize=\dtmp \epsffile{#2##1.eps}}
 294  \centerline{\vbox{\halign{####&\hfil##\hfil&##\cr #3\cr}}}\endgroup}
 295 \def\fgl[#1]#2\lgf{\begingroup\dtmp=\hsize \divide\dtmp by#1 \advance\dtmp-1em
 296   \def\lab[##1]{\hbox to\dtmp{\hfil##1\hfil}}
 297  \centerline{\vbox{\halign{####&\hfil##\hfil&##\cr #2\cr}}}\endgroup}
 298 % \Pp<lattice>:<Pc type>:<number\(<order>),...>:<Pc prob.>:<std. error>:
 299 \def\Pp#1:#2:#3:#4:#5:{\begingroup \def\(##1){({\bf##1})}
 300  \def\Vii{Vn 2d} \def\Viii{Vn 3d} \def\Viis{Vn (2,3)s}
 301  ${}^{\hbox{\sixit #1}}_{\scriptscriptstyle #3}p_#2$ $=$ $#4\pm #5$\endgroup}
 302 \def\X#1:#2:#3:#4:#5:{\begingroup \def\(##1){({\bf##1})} \def\Vii{Vn 2d} \def\Viii{Vn 3d}
 303  \def\Viis{Vn (2,3)s}
 304  \ifx0#5 ${}^{\hbox{\sixit #1}}_{\scriptscriptstyle #3}x_#2$ $=$ $#4$
 305   \else${}^{\hbox{\sixit #1}}_{\scriptscriptstyle #3}x_#2$ $=$ $#4\pm #5$\fi\endgroup}
 306 \def\din#1.#2.#3.#4.{\begingroup\def\der[##1]{{\bf ##1},} % #1 \latin, #2 \tenbf , #3 \ninepoint
 307  \def\w##1:{\dtmpi=\hsize \advance\dtmpi-.3em \dtmpii=\dtmpi \advance\dtmpii-.3em
 308   \parshape=2 0pt \dtmpi .3em \dtmpii {#2 ##1.~}}
 309  \def\Bf[##1]{{\bf##1}} \def\(##1){{\it##1}} \def\Sl[##1]{{\sl##1}} \def\[##1]{{\tt##1}}
 310  #1\dtmp=\hsize
 311  \begindoublecolumns\hsize=\dtmp \divide\hsize by2 \advance\hsize by-2em
 312  \baselineskip.9em\parindent0pt #3
 313  \input #4
 314  \enddoublecolumns\endgroup}
 315 \long\def\t[#1]{\def\next{#1}{\tt\frenchspacing\expandafter\strip\meaning\next}}
 316  \def\strip#1>{}
```

```
\def\idx[#1]{\beginindex\input #1\endindex}
\def\tchd{\def\leftheadline{\hbox to \pagewidth{%
    \vbox to 10pt{}% strut to position the baseline
    \llap{\tenbf\romannumeral\folio\kern1pc}% folio to left of text
    \tenit\rhead\hfil% running head flush left
    }}
  \def\rightheadline{\hbox to \pagewidth{%
    \vbox to 10pt{}% strut to position the baseline
    \hfil\tenit\rhead\/% running head flush right
    \rlap{\kern1pc\tenbf\romannumeral\folio}% folio to right of text
    }}}
\newbox\partialpage
\def\begindoublecolumns{\dtmp=\hsize \dtmpi=\vsize\begingroup
  \output={\global\setbox\partialpage=\vbox{\unvbox255\bigskip}}\eject
  \output={\doublecolumnout} \hsize=\dtmp \divide\hsize by2 \advance\hsize-.4em
  \vsize=\dtmpi \multiply\vsize by2 \advance\vsize by.4em}
\def\enddoublecolumns{\output={\balancecolumns}\eject
  \endgroup \pagegoal=\vsize}
\def\doublecolumnout{\splittopskip=\topskip \splitmaxdepth=\maxdepth
  \dtmp=\dtmpi \dimen@=\dtmp \advance\dimen@ by-\ht\partialpage
  \setbox0=\vsplit255 to\dimen@ \setbox2=\vsplit255 to\dimen@
  \onepageout\pagesofar
  \unvbox255 \penalty\outputpenalty}
\def\pagesofar{\unvbox\partialpage
  \wd0=\hsize \wd2=\hsize \hbox to\pagewidth{\box0\hfil\box2}}
\def\balancecolumns{\setbox0=\vbox{\unvbox255} \dimen@=\ht0
  \advance\dimen@ by\topskip \advance\dimen@ by-\baselineskip
  \divide\dimen@ by2 \splittopskip=\topskip
  {\vbadness=10000 \loop \global\setbox3=\copy0
    \global\setbox1=\vsplit3 to\dimen@
    \ifdim\ht3>\dimen@ \global\advance\dimen@ by1pt \repeat}
  \setbox0=\vbox to\dimen@{\unvbox1}
  \setbox2=\vbox to\dimen@{\unvbox3}
  \pagesofar}
\def\begintriplecolumns{\dtmp=\hsize \dtmpi=\vsize \begingroup
  \output={\global\setbox\partialpage=\vbox{\unvbox255\bigskip}}\eject
  \output={\triplecolumnout} \hsize=\dtmp \divide\hsize by3 \advance\hsize by-.4em
   \vsize=\dtmpi \multiply\vsize by3 \advance\vsize by.4em}
\def\endtriplecolumns{\output={\balancethreecolumns}\eject
  \endgroup \pagegoal=\vsize}
\def\triplecolumnout{\splittopskip=\topskip \splitmaxdepth=\maxdepth
  \dtmpii=\dtmpi \dimen@=\dtmpii \advance\dimen@ by-\ht\partialpage
  \setbox0=\vsplit255 to\dimen@ \setbox2=\vsplit255 to\dimen@ \setbox4=\vsplit255 to\dimen@
  \onepageout\pagesofar
  \unvbox255 \penalty\outputpenalty}
\def\pagesofar{\unvbox\partialpage
  \wd0=\hsize \wd2=\hsize \wd4=\hsize \hbox to\pagewidth{\box0\hfil\box2\hfil\box4}}
\def\balancethreecolumns{\setbox0=\vbox{\unvbox255} \dimen@=\ht0
  \advance\dimen@ by\topskip \advance\dimen@ by-\baselineskip
  \divide\dimen@ by3 \splittopskip=\topskip
  {\vbadness=10000 \loop \global\setbox5=\copy0
    \global\setbox1=\vsplit5 to\dimen@
    \global\setbox3=\vsplit5 to\dimen@
    \ifdim\ht5>\dimen@ \global\advance\dimen@ by1pt \repeat}
  \setbox0=\vbox to\dimen@{\unvbox1}
  \setbox2=\vbox to\dimen@{\unvbox3}
  \setbox4=\vbox to\dimen@{\unvbox5}
  \pagesofar}
% cf. manmac.tex
\def\onepageout#1{\shipout\vbox{ % here we define one page of output
    \offinterlineskip % butt the boxes together
    \vbox to 3pc{ % this part goes on top of the 44pc pages
      \iftitle % the next is used for title pages
        \global\titlefalse % reset the titlepage switch
        \setcornerrules % for camera alignment
      \else\ifodd\pageno \rightheadline\else\leftheadline\fi\fi
      \vfill} % this completes the \vbox to 3pc
    \vbox to \pageheight{\dtmp=\hsize \advance\dtmp by1pc
      \ifvoid\margin\else % marginal info is present
        \rlap{\kern\dtmp\vbox to\z@{\kern4pt\box\margin \vss}}\fi
      #1 % now insert the main information
      \ifvoid\footins\else % footnote info is present
        \vskip\skip\footins \kern-3pt
        \hrule height\ruleht width\pagewidth \kern-\ruleht \kern3pt
        \unvbox\footins\fi
      \boxmaxdepth=\maxdepth
      } % this completes the \vbox to \pageheight
    }
  \advancepageno}
\def\pap#1{\begingroup\parindent=1em \ninerm\input #1\endgroup}
\def\lhd[#1]{\leftline{\(\tenrm\bf #1)}}
\def\chd[#1]{\centerline{\(\tenrm #1)}}
```

```
\def\qte[#1][#2]{\begingroup\dtmp=\hsize \advance\dtmp-3em
 \setbox1=\vbox{\hsize=\dtmp \parindent=0pt #1\smallskip\rightline{#2}}
 \smallskip\centerline{\copy1}\smallskip\endgroup}
\def\quo[#1]{\begingroup \def\|{\par} \dtmp=\hsize \advance\dtmp-5em
 \setbox1=\vbox{\hsize=\dtmp \parindent=0pt #1}
 \smallskip\centerline{\copy1}\smallskip\endgroup}
\def\bitm[#1]{\smallskip\begingroup \def\itm{\item{#1}}}
\def\eitm{\endgroup\smallskip}
\def\btmn#1\etmn{\smallskip\begingroup \tmp=0
  \def\itm{\advance\tmp by 1 \item{\the\tmp.}} #1\endgroup\smallskip}
\def\btma#1\etma{\smallskip\begingroup \tmp=96
  \def\itm{\advance\tmp by 1 \item{\char\tmp.}} #1\endgroup\smallskip}
\def\btab[#1]#2\etab{\centerline{\setbox1=\vbox{#2}
 \vbox{\halign{#1\let|=&
 \let*=\cr
 \copy1}}}}
\def\tdim{\dtmp=\hsize \advance\dtmp by-2em \hsize=\dtmp}
\newcount\pam \pam=96
\def\lox(#1)[#2]{\centerline{{\twelvebf #1}}\global\advance\pam by 1
  \begingroup\medskip\begingroup\parindent=0pt\obeylines
   \immediate\write\ref{\string\ix[#2](\char\the\pam)}
   \immediate\write\ref{\string\ix[#2n](#1)} \immediate\write\ref{\string\ix[#2p](\number\pageno)}
   \ix[#1](\char\the\pam) \ix[#2n](#1) \ix[#2p](\number\pageno)
   \ifundefined{toc} \else\immediate\write\toc{\x[#2]. \x[#2n]\hfil\dotfill\x[#2p]}\fi
  \input #2
  \endgroup\endgroup\bigskip}
\def\loy(#1)[#2]{\centerline{{\twelvebf #1}}\global\advance\pam by 1
  \begingroup\medskip\begingroup\parindent=0pt\obeylines
   \immediate\write\ref{\string\ix[#2](\char\the\pam)}
   \immediate\write\ref{\string\ix[#2n](#1)} \immediate\write\ref{\string\ix[#2p](\number\pageno)}
   \ix[#1](\char\the\pam) \ix[#2n](#1) \ix[#2p](\number\pageno)
  \endgroup\endgroup\bigskip}
\newcount\apc \apc=64
\def\app[#1]#2:#3:{\advance\apc by 1
 \immediate\write\ref{\string\ix[#1](\char\the\apc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\char\the\apc) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \beginchapter {\Title\ Appendix} {}{\char\apc}. {\x[#1n]}\par
 \ifundefined{toc} \else\immediate\write\toc{\x[#1]. \x[#1n]\hfil\dotfill\x[#1p]}\fi
 \S\ \Bf \x[#1]. \x[#1n]:\par \begingroup\input #3 \endgroup\par
 \stn=0 \alc=0 \fgc=0 \tbc=0 \cac=0 \tmc=0 \asm=0 \dfc=0}
\newcount\cch
\def\chp[#1]#2:#3:{\advance\cch by 1
 \immediate\write\ref{\string\ix[#1](\the\cch)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \beginchapter {\Title\ Chapter} {}{\x[#1]}. {\x[#1n]}\par
 \ifundefined{toc} \else\immediate\write\toc{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi
 \S\ \Bf \x[#1]. \x[#1n]:\par \begingroup\input #3 \endgroup\par
 \stn=0 \alc=0 \fgc=0 \tbc=0 \cac=0 \tmc=0 \asm=0 \dfc=0}
\newcount\stn
\def\sap[#1]#2:#3:{\advance\stn by 1
 \immediate\write\ref{\string\ix[#1](\char\the\apc.\the\stn)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\char\the\apc.\the\stn) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \beginsection
 \ifundefined{toc} \else\immediate\write\toc{\q\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi
 \S\ {\bf\x[#1] \x[#1n]}\par \begingroup\input #3 \endgroup\par}
\def\sct[#1]#2:#3:{\advance\stn by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\stn)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\stn) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \beginsection
 \ifundefined{toc} \else\immediate\write\toc{\q\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi
 \S\ {\bf\x[#1] \x[#1n]}\par \begingroup\input #3 \endgroup\par}
\newcount\alc
\def\alg[#1]#2:{\advance\alc by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\alc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\alc) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \ifundefined{log} \else\immediate\write\log{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\newcount\eqc
\def\eqn{\global\advance\eqc by 1\eqno{(\bf\the\eqc)_{\hbox{\sevenrm\romannumeral\the\cch}}}}
\def\eql{\global\advance\eqc by 1 (\bf\the\eqc)_{\hbox{\sevenrm\romannumeral\the\cch}}}
\def\eqa[#1]#2:{\global\advance\eqc by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\eqc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\eqc) \ix[#1n](#2) \ix[#1p](\number\pageno)
  \eqno{(\bf\the\eqc)_{\hbox{\sevenrm\romannumeral\the\cch}}}}
\def\eqal[#1]#2:{\global\advance\eqc by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\eqc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
```

```
 \ix[#1](\the\cch.\the\eqc) \ix[#1n](#2) \ix[#1p](\number\pageno)
  (\bf\the\eqc)_{\hbox{\sevenrm\romannumeral\the\cch}}}
\newcount\fgc
\def\fig[#1]#2:{\global\advance\fgc by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\fgc)}
 \immediate\write\ref{\string\ix[#1n](#2)}
 \ifundefined{#1p}\immediate\write\ref{\string\ix[#1p](\number\pageno)} \ix[#1p](\number\pageno)\fi
 \ix[#1](\the\cch.\the\fgc) \ix[#1n](#2)
 \ifundefined{lof} \else\immediate\write\lof{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\newcount\tbc
\def\tab[#1]#2:{\advance\tbc by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\tbc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\tbc) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \ifundefined{lot} \else\immediate\write\lot{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\newcount\cac
\def\proclaim #1. #2\par{\medbreak
  \noindent{\bf#1.\enspace}{\sl#2\par}%
  \ifdim\lastskip<\medskipamount \removelastskip\penalty55\medskip\fi}
\def\cly[#1]#2:#3\par{\advance\cac  by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\cac)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\tmc[\the\cac]) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \proclaim Corollary \x[#1]. #3\par
 \ifundefined{lom} \else\immediate\write\lom{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\newcount\asm
\def\asu[#1]#2:#3\par{\advance\asm  by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\asm)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\tmc) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \proclaim Assumption \x[#1]. #3\par
 \ifundefined{los} \else\immediate\write\los{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\newcount\dfc
\def\dfn[#1]#2:#3\par{\advance\dfc  by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\dfc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\tmc) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \proclaim Definition \x[#1]. #3\par
 \ifundefined{lod} \else\immediate\write\lod{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\newcount\tmc
\def\thm[#1]#2:#3\par{\advance\tmc  by 1
 \immediate\write\ref{\string\ix[#1](\the\cch.\the\tmc)}
 \immediate\write\ref{\string\ix[#1n](#2)} \immediate\write\ref{\string\ix[#1p](\number\pageno)}
 \ix[#1](\the\cch.\the\tmc) \ix[#1n](#2) \ix[#1p](\number\pageno)
 \proclaim Theorem \x[#1]. #3\par
 \ifundefined{lom} \else\immediate\write\lom{\x[#1]. \x[#1n]\dotfill\x[#1p]}\fi}
\def\xp[#1]{\immediate\write\ref{\string\ix[#1](\number\pageno)} \ix[#1](\number\pageno)}
\def\mctb[#1:#2:#3:#4:#5:#6:#7:#8]{\smallskip\begingroup\dtmp=\hsize
  \advance\dtmp2em \divide\dtmp by7
\dtmpi=0pt
 \setbox1=\vbox{\hsize=\dtmp #3}
 \dtmpii=\ht1
 \ifdim\dtmpii>\dtmpi \dtmpi=\dtmpii\fi
 \setbox1=\vbox{\hsize=\dtmp #4}
 \dtmpii=\ht1
 \ifdim\dtmpii>\dtmpi \dtmpi=\dtmpii\fi
 \setbox1=\vbox{\hsize=\dtmp #5}
 \dtmpii=\ht1
 \ifdim\dtmpii>\dtmpi \dtmpi=\dtmpii\fi
 \setbox1=\vbox{\hsize=\dtmp #6}
 \dtmpii=\ht1
 \ifdim\dtmpii>\dtmpi \dtmpi=\dtmpii\fi
 \setbox1=\vbox{\hsize=\dtmp #7}
 \dtmpii=\ht1
 \ifdim\dtmpii>\dtmpi \dtmpi=\dtmpii\fi
 \setbox1=\vbox{\hsize=\dtmp #8}
 \dtmpii=\ht1
 \ifdim\dtmpii>\dtmpi \dtmpi=\dtmpii\fi
 \centerline{\vbox{\halign{##\hfil&##\hfil&##\hfil&##\hfil&##\hfil&##\hfil&##\hfil\cr
 \multispan{6}{\hfill#1\hfill}\cr\trule
  \vbox{\hsize=\dtmp\parindent=0pt~\(Cell)} &\vbox{\hsize=\dtmp\parindent=0pt~\(Bond)}
 &\vbox{\hsize=\dtmp\parindent=0pt~\(cell)} &\vbox{\hsize=\dtmp\parindent=0pt~\(bond)}
 &\vbox{\hsize=\dtmp\parindent=0pt~\(vertex)} &\vbox{\hsize=\dtmp\parindent=0pt~\(edge)}
\cr
  #2\cr
 \vbox to\dtmpi{\hsize=\dtmp\parindent=0pt #3\hfil\vfill}
 &\vbox to\dtmpi{\hsize=\dtmp\parindent=0pt #4\hfil\vfill}
 &\vbox to\dtmpi{\hsize=\dtmp\parindent=0pt #5\hfil\vfill}
 &\vbox to\dtmpi{\hsize=\dtmp\parindent=0pt #6\hfil\vfill}
 &\vbox to\dtmpi{\hsize=\dtmp\parindent=0pt #7\hfil\vfill}
 &\vbox to\dtmpi{\hsize=\dtmp\parindent=0pt #8\hfil\vfill}\cr}}}\endgroup}
\def\mvmm(#1;#2;#3,#4;#5,#6){\begingroup\sevenrm\setbox1=\hbox{#1$\pm$#2}
```

```
 562  \ifx0#4\def\tmpa{#3}\else\def\tmpa{#3\times 10^{#4}}\fi
 563  \ifx0#6\def\tmpb{#5}\else\def\tmpb{#5\times 10^{#6}}\fi
 564  $\mathrel{\mathop{\copy1}\limits_{\tmpa\atop\tmpb}}$\endgroup}
 565 \def\cn(#1,#2/#3){\begingroup\sevenrm
 566  \setbox1=\hbox{#1[#2]} \dtmp=\wd1
 567  \setbox2=\hbox{\fiverm#3}
 568  \ifdim\wd2>\dtmp \dtmp=\wd2\fi
 569  \setbox2=\hbox{\fiverm#3}
 570  $\mathop{\copy1}\limits_{\copy2}$\endgroup}
 571 % 2-homohedral tilings
 572 \def\hm#1.{\begingroup\def\(##1){{\rm ##1}} \ifcase#1\or$3_3[3^3]7_1[3^7]$\or$3_3[3^3]9_3[3^9]_\(II)$
 573  \or$4_4[3^4]8_4[3^8]$\or$3_3[3^3]8_2[3^8]$\or$3_3[3^3]9_3[3^9]_\(III)$\or$4_3[3^4]10_6[3^{10}]_\(I)$
 574  \or$3_3[3^3]9_3[3^9]_\(I)$\or$4_2[3^4]10_4[3^{10}]$\or$4_3[3^4]10_6[3^{10}]_\(II)$\or$4_2[3^4]8_2[3^8]$
 575  \or$4_3[3^4]8_3[3^8]_\(I)$\or$4_3[3^4]8_3[3^8]_\(II)$\or$4_3[3^4]8_3[3^8]_\(III)$
 576  \or$4_4[3^4]7_2[3^7]_\(II)$\or$3_3[3^3]12_6[3^{12}]$\or$4_4[3^4]7_2[3^7]_\(I)$\or$5_4[3^5]7_4[3^7]_\(I)$
 577  \or$3_1[4^3]5_1[4^5]_\(I)$\or$3_1[4^3]5_1[4^5]_\(II)$\or$5_3[3^5]8_6[3^8]_\(II)$
 578  \or$5_3[3^5]8_6[3^8]_\(III)$\or$5_2[3^5]12_{12}[3^{12}]$\or$5_4[3^5]7_4[3^7]_\(II)$
 579  \or$5_3[3^5]8_6[3^8]_\(I)$\or$4_2[3^4]12_6[3^{12}]$\or$4_2[3^4]18_{12}[3^{18}]$\or$5_3[3^5]7_3[3^7]_\(I)$
 580  \or$5_3[3^5]7_3[3^7]_\(II)$\or$3_2[4^3]5_2[4^5]_\(I)$\or$3_2[4^3]5_2[4^5]_\(II)$\or$3_3[4^3]5_3[4^5]$
 581  \or$3_1[4^3]6_2[4^6]_\(I)$\or$3_1[4^3]6_2[4^6]_\(II)$\or$3_2[4^3]6_4[4^6]$\or$3_3[4^3]6_6[4^6]$
 582  \or$3_1[4^3]8_4[4^8]$\or$3_1[5^3]4_2[5^4]_\(I)$\or$3_1[5^3]4_2[5^4]$\or$3_2[5^3]4_4[5^4]$\fi\endgroup}
 583 \def\kittix{$\raise.1em\hbox{$>$}$\kern-.2em\raise.118em\hbox{:}Kittix}
 584 \def\blr{\begingroup \dtmp=\hsize \advance\dtmp-1em \dtmpi=\dtmp
 585  \advance\dtmpi-1em \divide\dtmpi by2 \parindent0pt \def\stn{\smallskip}
 586  \def\b[##1]{\bf ##1} \def\[##1]{\it ##1}
 587  \def\l##1:##2:{\setbox0=\vbox{\hsize=\dtmpi\sl##1} \setbox1=\vbox{\hsize=\dtmpi##2}
 588  \ifnum\ht0>\ht1 \dtmpii=\ht0 \else \dtmpii=\ht1 \fi
 589  \centerline{\vbox to\dtmpii{\hsize=\dtmpi\sl##1}\hfil\vbox to\dtmpii{\hsize=\dtmpi##2}}}}
 590 \def\elr{\endgroup}
 591 \def\wra[#1][#2][#3]{\begingroup\baselineskip=2em\setbox1=\hbox{#3}$\aleph(\hbox{#1})=$
 592   $\bigl\{$[#2], (\copy1)$\bigr\}$\endgroup}
 593 \def\wrs[#1][#2][#3][#4]{$\aleph(\hbox{#1})=$ $\bigl\{$[#2], (#3), $\langle$#4$\rangle$
$\bigr\}$}
 594 \def\head#1{\hbox{} \vskip#1} \def\setcornerrules{} \def\inpt[#1]{\input #1}
 595 \def\beginchapter#1 #2#3. #4\par{\global\exno=0
 596   \subsecno=0
 597   \def\chapno{#2#3}
 598   \titlepage
 599   \def\\{ } % \\'s in the title will be treated as spaces
 600   \message{#1 #2#3:} % show the chapter title on the terminal
 601   \xdef\rhead{#1 #2#3: #4\unskip}
 602   {\def\TeX{T\kern-.2em\lower.5ex\hbox{E}\kern-.06em X}
 603     \def\MF{{\vbox to30pt{}\manual ()*+,-.*}}
 604     \def\\{#3}
 605     \ifx\empty\\ \ifodd\pageno \rightline{\inchhigh #2\kern-.04em}
 606      \else\leftline{\inchhigh #2\kern-.04em}\fi
 607     \else\ifodd\pageno \rightline{\inchhigh #2\kern-.06em#3\kern-.04em}
 608      \else\leftline{\inchhigh #2\kern-.04em}\fi\fi
 609     \vskip .75pc
 610     \baselineskip 16pt \lineskiplimit \titlelsl \lineskip 3pt
 611     \let\\=\cr % now the \\'s are line dividers
 612     \ifodd\pageno \halign{\line{\titlefont\hfil##}\\#4\unskip\\}
 613      \else\halign{\line{\titlefont ##\hfil}\\#4\unskip\\}\fi}
 614   \tenpoint
 615   \noindent\ignorespaces}
 616 \def\dct:#1:#2:#3:{\dtmp=\hsize \begindoublecolumns\hsize=\dtmp \divide\hsize by2
 617  \def\rhead{\firstmark\hfil{\tenit #1--#2 dictionary}\hfil\botmark}
 618  \def\leftheadline{\hbox to \pagewidth{\vbox to 10pt{}\llap{\tenbf\folio\kern1pc}\tenbf\rhead}}
 619  \def\rightheadline{\hbox to \pagewidth{\vbox to 10pt{}\tenbf\rhead\rlap{\kern1pc\tenbf\folio}}}
 620  \baselineskip.9em\parindent=0pt\eightpoint \input #3
 621  \enddoublecolumns\cut}
 622 \def\rightharpoonupfill{$\m@th\smash-\mkern-7mu%
 623   \cleaders\hbox{$\mkern-2mu\smash-\mkern-2mu$}\hfill
 624   \mkern-7mu\mathord\rightharpoonup$}
 625 \def\overrightharpoonup#1{\vbox{\m@th\ialign{##\crcr
 626       \rightharpoonupfill\crcr\noalign{\kern-\p@\nointerlineskip}
 627       $\hfil\displaystyle{#1}\hfil$\crcr}}}
 628 % \input grammar
 629 \input language
 630 \asl\bengali\chem\czech\daiy\deutsch\espanlol\francjais\gaelic\grammar\hindi\hungarian\lanna\latin\lating
 631 \math\money\nihongo\norge\pali\physics\polish\russian\sanskrit\serbo\slovak\vietnamese\zhongwen
 632 % edit index.tex -> dstidx.tex
 633 % 1. replace _! by ,
 634 % 2. append \begindoublecolumns\beginindex at top, and \endindex\enddoublecolumns at end
 635 % awk script
 636 % awk '{printf("%d %s\n",NR,$0);}' tmp > tmp1
 637 % awk '{for(i=1:i<=NR:i++){for(j=2:j<=NF:j++){printf("%s",$j)}}}' tmp > tmp1
 638 % \def\{$\aleph$}
 639 % sort ref.tex | awk '{printf("\\def\\%s\n",$0);}' > tmp1
```

§ A.12 Language macros for TEX

```
\def\asl{\def\uq.{\d u}
  \def\oq.{\d o}
  \def\uoq.{\d uo}
  \def\iq.{\d i}
  \def\eq.{\d e}
  \def\ieq.{\d ie}
  \def\aq.{\d a}
  \def\aoq.{\d ao}
  \def\oaq.{\d oa}
  \def\uiq.{\d ui}
  \def\iuq.{\d iu}
  \def\ouq.{\d ou}
  \def\ueq.{\d ue}
  \def\oeq.{\d oe}
  \def\aeq.{\d ae}
  \def\euq.{\d eu}
  \def\iaq.{\d ia}
  \def\aqch.{\d ach}
  \def\aqct.{\d act}
  \def\aqsp.{\d asp}
  \def\oqss.{\d oss}
  \def\eqnt.{\d ent}
  \def\iqde.{\d ide}
  \def\iqnt.{\d int}
 \def\th.{\‘t} \def\Th.{\‘T}
  \def\tsh.{\‘ts} \def\Tsh.{\‘Ts}
  \def\tzh.{\‘tz} \def\Tzh.{\‘Tz}
  \def\bh.{\‘b} \def\Bh.{\‘B}
  \def\bsh.{\‘bs} \def\Bsh.{\‘Bs}
  \def\sh.{\‘s} \def\Sh.{\‘S}
  \def\ch.{\‘c} \def\Ch.{\‘C}
  \def\dh.{\‘s} \def\Dh.{\‘D}
  \def\fh.{\‘f} \def\Fh.{\‘F}
  \def\gh.{\‘g} \def\Gh.{\‘G}
  \def\gdh.{\‘gd} \def\Gdh.{\‘Gd}
  \def\hh.{\‘h} \def\Hh.{\‘H}
  \def\jh.{\‘\j} \def\Jh.{\‘J}
  \def\kh.{\‘k} \def\Kh.{\‘K}
  \def\lh.{\‘l} \def\Lh.{\‘L}
  \def\zh.{\‘z} \def\Zh.{\‘Z}
  \def\ph.{\‘p} \def\Ph.{\‘P}
  \def\rh.{\‘r} \def\Rh.{\‘R}
  \def\vh.{\‘v} \def\Vh.{\‘V}
  \def\wh.{\‘w} \def\Wh.{\‘W}
  \def\xh.{\‘x} \def\Xh.{\‘X}
  \def\yh.{\‘y} \def\Yh.{\‘Y}
  \def\ysh.{\‘ys} \def\Ysh.{\‘Ys}
  \def\qh.{\‘q} \def\Qh.{\‘Q}
 \def\tz.{\’t} \def\Tz.{\’T}
  \def\tsz.{\’ts} \def\Tsz.{\’Ts}
  \def\tzz.{\’tz} \def\Tzz.{\’Tz}
  \def\bz.{\’b} \def\Bz.{\’B}
  \def\bsz.{\’bs} \def\Bsz.{\’Bs}
  \def\sz.{\’s} \def\Sz.{\’S}
  \def\cz.{\’c} \def\Cz.{\’C}
  \def\dz.{\’s} \def\Dz.{\’D}
  \def\fz.{\’f} \def\Fz.{\’F}
  \def\gz.{\’g} \def\Gz.{\’G}
  \def\gdz.{\’gd} \def\Gdz.{\’Gd}
  \def\hz.{\’h} \def\Hz.{\’H}
  \def\jz.{\’\j} \def\Jz.{\’J}
  \def\kz.{\’k} \def\Kz.{\’K}
  \def\lz.{\’l} \def\Lz.{\’L}
  \def\zz.{\’z} \def\Zz.{\’Z}
  \def\pz.{\’p} \def\Pz.{\’P}
  \def\rz.{\’r} \def\Rz.{\’R}
  \def\vz.{\’v} \def\Vz.{\’V}
  \def\wz.{\’w} \def\Wz.{\’W}
  \def\xz.{\’x} \def\Xz.{\’X}
  \def\yz.{\’y} \def\Yz.{\’Y}
  \def\ysz.{\’ys} \def\Ysz.{\’Ys}
  \def\qz.{\’q} \def\Qz.{\’Q}
 \def\ty.{\^t} \def\Ty.{\^T}
  \def\tsy.{\^ts} \def\Tsy.{\^Ts}
  \def\tyy.{\^ty} \def\Tyy.{\^Ty}
  \def\by.{\^b} \def\By.{\^B}
  \def\bsy.{\^bs} \def\Bsy.{\^Bs}
  \def\sy.{\^s} \def\Sy.{\^S}
  \def\cy.{\^c} \def\Cy.{\^C}
```

```
  \def\dy.{\^s} \def\Dy.{\^D}
  \def\fy.{\^f} \def\Fy.{\^F}
  \def\gy.{\^g} \def\Gy.{\^G}
  \def\gdy.{\^gd} \def\Gdy.{\^Gd}
  \def\hy.{\^h} \def\Hy.{\^H}
  \def\jy.{\^\j} \def\Jy.{\^J}
  \def\ky.{\^k} \def\Ky.{\^K}
  \def\ly.{\^l} \def\Ly.{\^L}
  \def\yy.{\^y} \def\Yy.{\^Y}
  \def\py.{\^p} \def\Py.{\^P}
  \def\ry.{\^r} \def\Ry.{\^R}
  \def\vy.{\^v} \def\Vy.{\^V}
  \def\wy.{\^w} \def\Wy.{\^W}
  \def\xy.{\^x} \def\Xy.{\^X}
  \def\yy.{\^y} \def\Yy.{\^Y}
  \def\ysy.{\^ys} \def\Ysy.{\^Ys}
  \def\qy.{\^q} \def\Qy.{\^Q}
 \def\tq.{\d t} \def\Tq.{\d T}
  \def\tsq.{\d ts} \def\Tsq.{\d Ts}
  \def\tqq.{\d tq} \def\Tqq.{\d Tq}
  \def\bq.{\d b} \def\Bq.{\d B}
  \def\bsq.{\d bs} \def\Bsq.{\d Bs}
  \def\sq.{\d s} \def\Sq.{\d S}
  \def\cq.{\d c} \def\Cq.{\d C}
  \def\dq.{\d s} \def\Dq.{\d D}
  \def\fq.{\d f} \def\Fq.{\d F}
  \def\gq.{\d g} \def\Gq.{\d G}
  \def\gdq.{\d gd} \def\Gdq.{\d Gd}
  \def\hq.{\d h} \def\Hq.{\d H}
  \def\jq.{\d \j} \def\Jq.{\d J}
  \def\kq.{\d k} \def\Kq.{\d K}
  \def\lq.{\d l} \def\Lq.{\d L}
  \def\qq.{\d q} \def\Qq.{\d Q}
  \def\pq.{\d p} \def\Pq.{\d P}
  \def\rq.{\d r} \def\Rq.{\d R}
  \def\vq.{\d v} \def\Vq.{\d V}
  \def\wq.{\d w} \def\Wq.{\d W}
  \def\xq.{\d x} \def\Xq.{\d X}
  \def\wq.{\d w}
  \def\yq.{\d y} \def\Yq.{\d Y}
  \def\ysq.{\d ys} \def\Ysq.{\d Ys}
  \def\qq.{\d q} \def\Qq.{\d Q}
 \def\byh.{\`by} \def\Byh.{\`By}
  \def\tqy.{\d{\^t} \def\Tqy.{\d{\^T}}}
 \def\cniz.{cn\'\i} \def\Cniz.{Cn\'I}
  \def\mfiz.{mf\'\i} \def\Mfiz.{Mf\'I}
  \def\myah.{my\`a} \def\Myah.{My\`a}
  \def\myaz.{my\'a} \def\Myaz.{My\'a}
 \def\iqde.{\d ide}
 \def\vzl.{\'vl} \def\Vzl.{\'Vl}
}
\def\bengali{\def\aa.{\=a} \def\Aa.{\=A}
 \def\ii.{\=\i} \def\Ii.{\=I}
 \def\uu.{\=u} \def\Uu.{\=U}
 \def\rq.{\d r} \def\Rq.{\d R}
 \def\mq.{\d m} \def\Mq.{\d M}
 \def\hq.{\d h} \def\Hq.{\d H}
 \def\nx.{\.n} \def\Nx.{\.N}
 \def\nl.{\~n} \def\Nl.{\~N}
 \def\tq.{\d t} \def\Tq.{\d T}
 \def\tqh.{\d th} \def\Tqh.{\d Th}
 \def\dq.{\d d} \def\Dq.{\d D}
 \def\dqh.{\d dh} \def\Dqh.{\d Dh}
 \def\nq.{\d n} \def\Nq.{\d N}
 \def\sz.{\'s} \def\Sz.{\'S}
 \def\sq.{\d s} \def\Sq.{\d S}}
\def\chem{\def\fm[##1]{$\rm ##1$}
 \def\dgc{\hbox{$^\circ\hbox{C}$}}}
\def\czech{\def\cz.{\'c} \def\Cz.{\'C}
 \def\yz.{\'y} \def\Yz.{\'Y}}
\def\daiy{\def\av.{\v a} \def\Av.{\v A}
  \def\ah.{\`a} \def\Ah.{\`A}
  \def\af.{\=a} \def\Af.{\=A}
  \def\az.{\'a} \def\Az.{\'A}
 \def\aq.{\d a} \def\Aq.{\d A}
  \def\aqv.{\d{\v a}} \def\Aqv.{\d{\v A}}
  \def\aqh.{\d{\`a}} \def\Aqh.{\d{\`A}}
  \def\aqf.{\d{\=a}} \def\Aqf.{\d{\=A}}
  \def\aqz.{\d{\'a}} \def\Aqz.{\d{\'A}}
 \def\iv.{\v\i} \def\Iv.{\v I}
  \def\ih.{\`\i} \def\Ih.{\`I}
  \def\if.{\=\i} \def\If.{\=I}
```

```
 \def\iz.{\'\i} \def\Iz.{\'I}
\def\iq.{\d i} \def\Iq.{\d I}
 \def\iqv.{\d{\v\i}} \def\Iqv.{\d{\v I}}
 \def\iqh.{\d{\`\i}} \def\Iqh.{\d{\`I}}
 \def\iqf.{\d{\=\i}} \def\Iqf.{\d{\=I}}
 \def\iqz.{\d{\'\i}} \def\Iqz.{\d{\'I}}
\def\uev.{\v ue} \def\Uev.{\v Ue}
 \def\ueh.{\`ue} \def\Ueh.{\`Ue}
 \def\uef.{\=ue} \def\Uef.{\=Ue}
 \def\uez.{\'ue} \def\Uez.{\'Ue}
\def\uqe.{\d ue} \def\Uqe.{\d Ue}
 \def\uqev.{\d{\v u}e} \def\Uqev.{\d{\v U}e}
 \def\uqeh.{\d{\`u}e} \def\Uqeh.{\d{\`U}e}
 \def\uqef.{\d{\=u}e} \def\Uqef.{\d{\=U}e}
 \def\uqez.{\d{\'u}e} \def\Uqez.{\d{\'U}e}
\def\uv.{\v u} \def\Uv.{\v U}
 \def\uh.{\`u} \def\Uh.{\`U}
 \def\uf.{\=u} \def\Uf.{\=U}
 \def\uz.{\'u} \def\Uz.{\'U}
\def\uq.{\d u} \def\Uq.{\d U}
 \def\uqv.{\d{\v u}} \def\Uqv.{\d{\v U}}
 \def\uqh.{\d{\`u}} \def\Uqh.{\d{\`U}}
 \def\uqf.{\d{\=u}} \def\Uqf.{\d{\=U}}
 \def\uqz.{\d{\'u}} \def\Uqz.{\d{\'U}}
\def\ev.{\v e} \def\Ev.{\v E}
 \def\eh.{\`e} \def\Eh.{\`E}
 \def\ef.{\=e} \def\Ef.{\=E}
 \def\ez.{\'e} \def\Ez.{\'E}
\def\eq.{\d e} \def\Eq.{\d E}
 \def\eqv.{\d{\v e}} \def\Eqv.{\d{\v E}}
 \def\eqh.{\d{\`e}} \def\Eqh.{\d{\`E}}
 \def\eqf.{\d{\=e}} \def\Eqf.{\d{\=E}}
 \def\eqz.{\d{\'e}} \def\Eqz.{\d{\'E}}
\def\aev.{\v ae} \def\Aev.{\v Ae}
 \def\aeh.{\`ae} \def\Aeh.{\`Ae}
 \def\aef.{\=ae} \def\Aef.{\=Ae}
 \def\aez.{\'ae} \def\Aez.{\'Ae}
\def\aqe.{\d ae} \def\Aqe.{\d Ae}
 \def\aqev.{\d{\v a}e} \def\Aqev.{\d{\v A}e}
 \def\aqeh.{\d{\`a}e} \def\Aqeh.{\d{\`A}e}
 \def\aqef.{\d{\=a}e} \def\Aqef.{\d{\=A}e}
 \def\aqez.{\d{\'a}e} \def\Aqez.{\d{\'A}e}
\def\ov.{\v o} \def\Ov.{\v O}
 \def\oh.{\`o} \def\Oh.{\`O}
 \def\of.{\=o} \def\Of.{\=O}
 \def\oz.{\'o} \def\Oz.{\'O}
\def\oq.{\d o} \def\Oq.{\d O}
 \def\oqv.{\d{\v o}} \def\Oqv.{\d{\v O}}
 \def\oqh.{\d{\`o}} \def\Oqh.{\d{\`O}}
 \def\oqf.{\d{\=o}} \def\Oqf.{\d{\=O}}
 \def\oqz.{\d{\'o}} \def\Oqz.{\d{\'o}}
\def\auv.{\v au} \def\Auv.{\v Au}
 \def\auh.{\`au} \def\Auh.{\`Au}
 \def\auf.{\=au} \def\Auf.{\=Au}
 \def\auz.{\'au} \def\Auz.{\'Au}
\def\aqu.{\d au} \def\Aqu.{\d Au}
 \def\aquv.{\d{\v a}u} \def\Aquv.{\d{\v A}u}
 \def\aquh.{\d{\`a}u} \def\Aquh.{\d{\`A}u}
 \def\aquf.{\d{\=a}u} \def\Aquf.{\d{\=A}u}
 \def\aquz.{\d{\'a}u} \def\Aquz.{\d{\'A}u}
\def\oev.{\v oe} \def\Oev.{\v Oe}
 \def\oeh.{\`oe} \def\Oeh.{\`Oe}
 \def\oef.{\=oe} \def\Oef.{\=Oe}
 \def\oez.{\'oe} \def\Oez.{\'Oe}
\def\oqe.{\d oe} \def\Oqe.{\d Oe}
 \def\oqev.{\d{\v o}e} \def\Oqev.{\d{\v O}e}
 \def\oqeh.{\d{\`o}e} \def\Oqeh.{\d{\`O}e}
 \def\oqef.{\d{\=o}e} \def\Oqef.{\d{\=O}e}
 \def\oqez.{\d{\'o}e} \def\Oqez.{\d{\'O}e} \def\oqeez.{\d{\'o}ee}
\def\iav.{\v\i a} \def\Iav.{\v Ia}
 \def\iah.{\`\i a} \def\Iah.{\`Ia}
 \def\iaf.{\=\i a} \def\Iaf.{\=Ia}
 \def\iaz.{\'\i a} \def\iaz.{\'\i a}
\def\iqa.{\d ia} \def\Iqa.{\d Ia}
 \def\iqav.{\d{\v\i}a} \def\Iqav.{\d{\v I}a}
 \def\iqah.{\d{\`\i}a} \def\Iqah.{\d{\`I}a}
 \def\iqaf.{\d{\=\i}a} \def\Iqaf.{\d{\=I}a}
 \def\iqaz.{\d{\'\i}a} \def\Iqaz.{\d{\'I}a}
\def\uav.{\v ua} \def\Uav.{\v Ua}
 \def\uah.{\`ua} \def\Uah.{\`Ua}
 \def\uaf.{\=ua} \def\Uaf.{\=Ua}
 \def\uaz.{\'ua} \def\Uaz.{\'Ua}
```

```
\def\uqa.{\d ua} \def\Uqa.{\d Ua}
 \def\uqav.{\d{\v u}a} \def\Uqav.{\d{\v U}a}
 \def\uqah.{\d{\`u}a} \def\Uqah.{\d{\`U}a}
 \def\uqaf.{\d{\=u}a} \def\Uqaf.{\d{\=U}a}
 \def\uqaz.{\d{\'u}a} \def\Uqaz.{\d{\'U}a}
\def\oav.{\v oa} \def\Oav.{\v Oa}
 \def\oah.{\`oa} \def\Oah.{\`Oa}
 \def\oaf.{\=oa} \def\Oaf.{\=Oa}
 \def\oaz.{\'oa} \def\Oaz.{\'Oa}
\def\oqa.{\d oa} \def\Oqa.{\d Oa}
 \def\oqav.{\d{\v o}a} \def\Oqav.{\d{\v O}a}
 \def\oqah.{\d{\`o}a} \def\Oqah.{\d{\`O}a}
 \def\oqaf.{\d{\=o}a} \def\Oqah.{\d{\=O}a}
 \def\oqaz.{\d{\'o}a} \def\Oqaz.{\d{\'O}a}
\def\amq.{a\d m} \def\Amq.{A\d m}
 \def\amqv.{\v a\d m} \def\Amqv.{\v A\d m}
 \def\amqh.{\`a\d m} \def\Amqh.{\`A\d m} \def\ammqh.{\`a\d m\d{\"m}}
 \def\amqf.{\=a\d m} \def\Amqf.{\=A\d m}
 \def\amqz.{\'a\d m} \def\Amqz.{\'A\d m}
\def\aiv.{\v ai} \def\aiv.{\v ai}
 \def\aih.{\`ai} \def\aih.{\`ai}
 \def\aif.{\=ai} \def\aif.{\=ai}
 \def\aiz.{\'ai} \def\aiz.{\'ai}
\def\aiq.{a\d i} \def\Aqi.{\d Ai}
 \def\aiqv.{\v a\d i} \def\Aiqv.{\v A\d i}
 \def\aiqh.{\`a\d i} \def\Aiqh.{\`A\d i}
 \def\aiqf.{\=a\d i} \def\Aiqf.{\=A\d i}
 \def\aiqz.{\'a\d i} \def\Aiqz.{\'A\d i}
\def\aov.{\v ao} \def\Aov.{\v Ao}
 \def\aoh.{\`ao} \def\Aoh.{\`Ao}
 \def\aof.{\=ao} \def\Aof.{\=Ao}
 \def\aoz.{\'ao} \def\Aoz.{\'Ao}
\def\auqh.{\`a\d u} \def\Auqh.{\`A\d u}
\def\tq.{\d t} \def\Tq.{\d T}
 \def\tqh.{\d th} \def\Tqh.{\d Th}
 \def\kqh.{\d kh} \def\Kqh.{\d Kh}
 \def\gq.{\d g} \def\Gq.{\d G}
 \def\dq.{\d d} \def\Dq.{\d D}
 \def\dz.{\'d} \def\Dz.{\'D}
 \def\dqz.{\d{\'d}} \def\Dqz.{\d{\'D}}
 \def\dqh.{\d dh} \def\Dqh.{\d Dh}
 \def\nq.{\d n} \def\Nq.{\d N}
 \def\bq.{\d b} \def\Bq.{\d B}
 \def\fq.{\d f} \def\Fq.{\d F}
 \def\hq.{\d h} \def\Hq.{\d H}
 \def\lqq.{\d l} \def\Lq.{\d L}
 \def\rq.{\d r} \def\Rq.{\d R}
 \def\sq.{\d s} \def\Sq.{\d S}
 \def\yq.{\d y} \def\Yq.{\d Y}
 \def\lz.{\'l} \def\Lz.{\'L}
 \def\sz.{\'s} \def\Sz.{\'S}
\def\ae.{\"a}
 \def\bqe.{\d{\"b}}
 \def\nge.{\"ng}
 \def\ke.{\"k}
 \def\ne.{\"n}
 \def\ye.{\"y}
\def\ax.{\.a} \def\Ax.{\.A}
 \def\aux.{\.au}
 \def\aux.{\d{\.a}u} \def\Aux.{\d{\.A}u}
 \def\bx.{\.b}
 \def\bhx.{\.bh}
 \def\cx.{\.c}
 \def\dx.{\.d}
 \def\dhx.{\.dh}
 \def\dqhx.{\d{\.dh}}
 \def\dzx.{\'dx}
 \def\fx.{\.f}
 \def\fqx.{\d{\.f}}
 \def\gx.{\.g}
 \def\ghx.{\.gh}
 \def\hx.{\.h}
 \def\hqx.{\d{\.h}}
 \def\kx.{\.k}
 \def\khx.{\.kh}
 \def\lx.{\.l}
 \def\mx.{\.m}
 \def\nx.{\.n}
 \def\nqx.{\d{\.n}}
 \def\px.{\.p}
 \def\rx.{\.r}
 \def\sx.{\.s}
```

```
  \def\sqx.{\d{\.s}}
  \def\szx.{\d{\'s}}
  \def\tx.{\.t}
  \def\thx.{\.th}
  \def\tqx.{\d{\.t}}
  \def\tqhx.{\d{\.th}}
  \def\vx.{\.v}
  \def\yx.{\.y}
  \def\yqx.{\d{\.y}}
  \def\zx.{\.z}}

\def\deutsch{\def\acc{{\it acc}}
  \def\dat{{\it dat}}
  \def\eie{{\it e-e}}
  \def\eim{{\it e-m}}
  \def\eir{{\it e-r}}
  \def\etw{{\it et}}
  \def\gen{{\it gen}}
  \def\jm{{\it jm}}
  \def\jn{{\it jn}}
  \def\ohn{{\it ohn.}}
  \def\Ae.{\"A} \def\ae.{\"a}
  \def\Oe.{\"O} \def\oe.{\"o}
  \def\Ue.{\"U} \def\ue.{\"u}
  \def\Ss.{\ss }}
\def\espanlol{\def\nl.{\~n} \def\Nl.{\~N}}
\def\francjais{\def\qch{{\it q.ch.}}
 \def\Az.{\'A} \def\az.{\'a}
  \def\Ez.{\'E} \def\ez.{\'e}
 \def\Ah.{\`A} \def\ah.{\`a}
  \def\Eh.{\`E} \def\eh.{\`e}
 \def\Ey.{\^E} \def\ey.{\^e}
  \def\Iy.{\^I} \def\iy.{\^\i}
 \def\Ie.{\"I} \def\ie.{\"\i}}
\def\gaelic{\def\ah.{\`a} \def\Ah.{\'A}
 \def\eh.{\`e} \def\Eh.{\'E}
 \def\ih.{\`\i} \def\Ih.{\`I}
 \def\oh.{\`o} \def\Oh.{\`O}
 \def\uh.{\`u} \def\Uh.{\`U}}
\def\grammar{\def\ab{{\it abrv}}
 \def\adv{{\it adv}}
 \def\adj{{\it adj}}
 \def\ant{{\it ant}}    % antonym
 \def\arcc{{\it archaic}}
 \def\atb{{\it atrib}}  % attributive noun
 \def\auxi{{\it aux}}    % auxiliary
 \def\brit{{\it Brit.}}  % British
 \def\cca{{\it circa}}
 \def\cf{{\it cf}}
 \def\chem{{\it chem}}  % chemistry
 \def\chss{{\it chess}} % chess
 \def\cj{{\it conj}}    % conjunction
 \def\clfr{{\it classf}}% classifier
 \def\col{{\it col}}     % colloquial
 \def\cor{{\it corrupt}} % corrupted
 \def\dd{$\ldots$}
 \def\de{{\it derog}}   % derogatory
 \def\det{{\it det}}    % determiner
 \def\dto{{\it ditto}}
 \def\dy{{\it dy}}      % daiy
 \def\eg{{\it eg}}      % exempli gratia
 \def\esp{{\it esp}}
 \def\etc{{\it etc}}
 \def\etl{{\it et al}}
 \def\expr{{\it expr}}    % expression
 \def\f{{\it f}}        % feminine
 \def\figu.{{\it fig}}    % figurative
 \def\fml{{\it fml}}     % formal
 \def\Fr{{\it Fr.}}      % french
 \def\ger{{\it Ger.}}    % german
 \def\gram{{\it gram}}
 \def\grk{{\it Gr.}}
 \def\humou{{\it humour.}}
 \def\ie{{\it i.e.}} \def\idst{{\it i.e.}} % id est
 \def\ifm{{\it infml}}  % informal
 \def\inj{{\it intj}}   % interjection
 \def\joc{{\it joc}}    % jocular
 \def\lan{{\it Lan.}}    % lanna
 \def\lat{{\it Lat}.}    % latin
 \def\law{{\it law}}    % law jargon
 \def\lo{{\it loc}}     % local
 \def\lt{{\it lit}}     % literally
```

```
\def\m{{\it m}}        % masculine
\def\mat{{\it math}}
\def\matl{{\it mat}}   % material science
\def\med{{\it med}}    % medicine
\def\mil{{\it mil}}    % military
\def\mo{{\it mod}}     % modal
\def\mt{{\it m\d oay}} % moaydaiy
\def\n{{\it n}}        % noun
\def\nt{{\it nt}}      % neutrum
\def\nym{{\it acronym}}
\def\ob{{\it obs}}     % obsolete
\def\on{{\it ono}}     % onomatopoeia
\def\os{{\it o.s.}}
\def\prp{{\it prp}}
\def\pa{{\it Pa.}}      % pali
\def\part{{\it part}}  % participle
\def\pl{{\it pl}}      % plural
\def\pn{{\it pn}}      % pronoun
\def\ppt{{\it pres. part.}}   % present participle
\def\pres{{\it pres}}  % present
\def\prs{{\it prs}}    % prose
\def\prf{{\it prf}}    % prefix
\def\qed{{\it q.e.d.}}  % quod erat demonstrandum, which was to be proved
\def\qntf{{\it quantf}}% quantifier
\def\rad{{\it rad}}    % radical
\def\refl{{\it refl}}  % reflexive
\def\rh{{\it rhe}}     % rhetoric
\def\rp{\lower4pt\hbox{{\tenrm\~{}}}\raise4pt\hbox{}}
\def\sb{{\it sb}}
\def\sfx{{\it sfx}}
\def\sic{{\ninepoint [sic]}}
\def\sk{{\it Sk.}}    % sanskrit
\def\slng{{\it sl}}    % slang
\def\sng{{\it sing}}
\def\st{{\it sth}}
\def\dai{{\it Th.}}     % thai
\def\theor{{\it theor}}% theoretically
\def\us{{\it usu}}
\def\vb{{\it v}}
\def\viz{{\it viz.}} % videlicet
\def\vs{{\it vs}}
\def\vul{{\it vul}}    % vulgar
}
\def\hindi{\def\aa.{\=a} \def\Aa.{\=A}
 \def\ii.{\=\i} \def\Ii.{\=I}
 \def\uu.{\=u} \def\Uu.{\=U}
 \def\rq.{\d r} \def\Rq.{\d R}
 \def\nx.{\.n} \def\Nx.{\.N}
 \def\nl.{\~n} \def\Nl.{\~N}
 \def\nq.{\d n} \def\Nq.{\d N}
 \def\sz.{\'s} \def\Sz.{\'S}
 \def\sq.{\d s} \def\Sq.{\d S}
 \def\kq.{\d k} \def\Kq.{\d K}
 \def\kqh.{\d kh} \def\Kqh.{\d Kh}
 \def\gq.{\d g} \def\Gq.{\d G}
 \def\rq.{\d r} \def\Rq.{\d R}
 \def\rqh.{\d rh} \def\Rqh.{\d Rh}}
\def\hungarian{\def\az.{\'a} \def\Az.{\'A}
 \def\ez.{\'e} \def\Ez.{\'E}
 \def\iz.{\'\i} \def\Iz.{\'I}
 \def\oe.{\"o} \def\Oe.{\"O}
 \def\ow.{\H o} \def\Ow.{\H O}
 \def\uz.{\'u} \def\Uz.{\'U}
 \def\ue.{\"u} \def\Ue.{\"U}
 \def\uw.{\H u} \def\Uw.{\H U}}
\def\lanna{\def\av.{\v a} \def\Av.{\v A}
  \def\ah.{\`a} \def\Ah.{\`A}
  \def\ay.{\^a} \def\Ay.{\^A}
  \def\af.{\=a} \def\Af.{\=A}
  \def\az.{\'a} \def\Az.{\'A}
 \def\aq.{\d a} \def\Aq.{\d A}
  \def\aqv.{\d{\v a}} \def\Aqv.{\d{\v A}}
  \def\aqh.{\d{\`a}} \def\Aqh.{\d{\`A}}
  \def\aqy.{\d{\^a}} \def\Aqy.{\d{\^A}}
  \def\aqf.{\d{\=a}} \def\Aqf.{\d{\=A}}
  \def\aqz.{\d{\'a}} \def\Aqz.{\d{\'A}}
 \def\iv.{\v\i} \def\Iv.{\v I}
  \def\ih.{\`\i} \def\Ih.{\`I}
  \def\iy.{\^\i} \def\Iy.{\^I}
  \def\if.{\=\i} \def\If.{\=I}
  \def\iz.{\'\i} \def\Iz.{\'I}
 \def\iq.{\d i} \def\Iq.{\d I}
```

```
 \def\iqv.{\d{\v\i}} \def\Iqv.{\d{\v I}}
 \def\iqh.{\d{\‘\i}} \def\Iqh.{\d{\‘I}}
 \def\iqy.{\d{\^\i}} \def\Iqy.{\d{\^I}}
 \def\iqf.{\d{\=\i}} \def\Iqf.{\d{\=I}}
 \def\iqz.{\d{\’\i}} \def\Iqz.{\d{\’I}}
\def\uev.{\v ue} \def\Uev.{\v Ue}
 \def\ueh.{\‘ue} \def\Ueh.{\‘Ue}
 \def\uey.{\^ue} \def\Uey.{\^Ue}
 \def\uef.{\=ue} \def\Uef.{\=Ue}
 \def\uez.{\’ue} \def\Uez.{\’Ue}
\def\uqe.{\d ue} \def\Uqe.{\d Ue}
 \def\uqev.{\d{\v u}e} \def\Uqev.{\d{\v U}e}
 \def\uqeh.{\d{\‘u}e} \def\Uqeh.{\d{\‘U}e}
 \def\uqey.{\d{\^u}e} \def\Uqey.{\d{\^U}e}
 \def\uqef.{\d{\=u}e} \def\Uqef.{\d{\=U}e}
 \def\uqez.{\d{\’u}e} \def\Uqez.{\d{\’U}e}
\def\uv.{\v u} \def\Uv.{\v U}
 \def\uh.{\‘u} \def\Uh.{\‘U}
 \def\uy.{\^u} \def\Uy.{\^U}
 \def\uf.{\=u} \def\Uf.{\=U}
 \def\uz.{\’u} \def\Uz.{\’U}
\def\uq.{\d u} \def\Uq.{\d U}
 \def\uqv.{\d{\v u}} \def\Uqv.{\d{\v U}}
 \def\uqh.{\d{\‘u}} \def\Uqh.{\d{\‘U}}
 \def\uqy.{\d{\^u}} \def\Uqy.{\d{\^U}}
 \def\uqf.{\d{\=u}} \def\Uqf.{\d{\=U}}
 \def\uqz.{\d{\’u}} \def\Uqz.{\d{\’U}}
\def\ev.{\v e} \def\Ev.{\v E}
 \def\eh.{\‘e} \def\Eh.{\‘E}
 \def\ey.{\^e} \def\Ey.{\^E}
 \def\ef.{\=e} \def\Ef.{\=E}
 \def\ez.{\’e} \def\Ez.{\’E}
\def\eq.{\d e} \def\Eq.{\d E}
 \def\eqv.{\d{\v e}} \def\Eqv.{\d{\v E}}
 \def\eqh.{\d{\‘e}} \def\Eqh.{\d{\‘E}}
 \def\eqy.{\d{\^e}} \def\Eqy.{\d{\^E}}
 \def\eqf.{\d{\=e}} \def\Eqf.{\d{\=E}}
 \def\eqz.{\d{\’e}} \def\Eqz.{\d{\’E}}
\def\aev.{\v ae} \def\Aev.{\v Ae}
 \def\aeh.{\‘ae} \def\Aeh.{\‘Ae}
 \def\aey.{\^ae} \def\Aey.{\^Ae}
 \def\aef.{\=ae} \def\Aef.{\=Ae}
 \def\aez.{\’ae} \def\Aez.{\’Ae}
\def\aqe.{\d ae} \def\Aqe.{\d Ae}
 \def\aqev.{\d{\v a}e} \def\Aqev.{\d{\v A}e}
 \def\aqeh.{\d{\‘a}e} \def\Aqeh.{\d{\‘A}e}
 \def\aqey.{\d{\^a}e} \def\Aqey.{\d{\^A}e}
 \def\aqef.{\d{\=a}e} \def\Aqef.{\d{\=A}e}
 \def\aqez.{\d{\’a}e} \def\Aqez.{\d{\’A}e}
\def\ov.{\v o} \def\Ov.{\v O}
 \def\oh.{\‘o} \def\Oh.{\‘O}
 \def\oy.{\^o} \def\Oy.{\^O}
 \def\of.{\=o} \def\Of.{\=O}
 \def\oz.{\’o} \def\Oz.{\’O}
\def\oq.{\d o} \def\Oq.{\d O}
 \def\oqv.{\d{\v o}} \def\Oqv.{\d{\v O}}
 \def\oqh.{\d{\‘o}} \def\Oqh.{\d{\‘O}}
 \def\oqy.{\d{\^o}} \def\Oqy.{\d{\^O}}
 \def\oqf.{\d{\=o}} \def\Oqf.{\d{\=O}}
 \def\oqz.{\d{\’o}} \def\oqz.{\d{\’o}}
\def\ouv.{\v ou} \def\Ouv.{\v Ou}
 \def\ouh.{\‘ou} \def\Ouh.{\‘Ou}
 \def\ouy.{\^ou} \def\Ouy.{\^Ou}
 \def\ouf.{\=ou} \def\Ouf.{\=Ou}
 \def\ouz.{\’ou} \def\Ouz.{\’Ou}
\def\ouq.{o\d u} \def\Ouq.{O\d u}
 \def\ouqv.{\v o\d u} \def\Ouqv.{\v O\d u}
 \def\ouqh.{\‘o\d u} \def\Ouqh.{\‘O\d u}
 \def\ouqy.{\^o\d u} \def\Ouqy.{\^O\d u}
 \def\ouqf.{\=o\d u} \def\Ouqf.{\=O\d u}
 \def\ouqz.{\’o\d u} \def\ouqz.{\’o\d u}
\def\oqu.{\d ou} \def\Oqu.{\d Ou}
 \def\oquv.{\d{\v o}u} \def\Oquv.{\d{\v O}u}
 \def\oquh.{\d{\‘o}u} \def\Oquh.{\d{\‘O}u}
 \def\oquy.{\d{\^o}u} \def\Oquy.{\d{\^O}u}
 \def\oquf.{\d{\=o}u} \def\Oquf.{\d{\=O}u}
 \def\oquz.{\d{\’o}} \def\oquz.{\d{\’o}u}
\def\auv.{\v au} \def\Auv.{\v Au}
 \def\auh.{\‘au} \def\Auh.{\‘Au}
 \def\auy.{\^au} \def\Auy.{\^Au}
 \def\auf.{\=au} \def\Auf.{\=Au}
 \def\auz.{\’au} \def\Auz.{\’Au}
```

```
\def\auq.{a\d u} \def\Auq.{A\d u}
 \def\auqv.{\v a\d u} \def\Auqv.{\v A\d u}
 \def\auqh.{\`a\d u} \def\Auqh.{\`A\d u}
 \def\auqy.{\^a\d u} \def\Auqy.{\^A\d u}
 \def\auqf.{\=a\d u} \def\Auqf.{\=A\d u}
 \def\auqz.{\'a\d u} \def\Auqz.{\'A\d u}
\def\aqu.{\d au} \def\Aqu.{\d Au}
 \def\aquv.{\d{\v a}u} \def\Aquv.{\d{\v A}u}
 \def\aquh.{\d{\`a}u} \def\Aquh.{\d{\`A}u}
 \def\aquy.{\d{\^a}u} \def\Aquy.{\d{\^A}u}
 \def\aquf.{\d{\=a}u} \def\Aquf.{\d{\=A}u}
 \def\aquz.{\d{\'a}u} \def\Aquz.{\d{\'A}u}
\def\aiv.{\v ai} \def\aiv.{\v ai}
 \def\aih.{\`ai} \def\aih.{\`ai}
 \def\aiy.{\^ai} \def\aiy.{\^ai}
 \def\aif.{\=ai} \def\aif.{\=ai}
 \def\aiz.{\'ai} \def\aiz.{\'ai}
\def\aov.{\v ao} \def\Aov.{\v Ao}
 \def\aoh.{\`ao} \def\Aoh.{\`Ao}
 \def\aoy.{\^ao} \def\Aoy.{\^Ao}
 \def\aof.{\=ao} \def\Aof.{\=Ao}
 \def\aoz.{\'ao} \def\Aoz.{\'Ao}
\def\amq.{a\d m} \def\Amq.{A\d m}
 \def\amqv.{\v a\d m} \def\Amqv.{\v A\d m}
 \def\amqh.{\`a\d m} \def\Amqh.{\`A\d m}
 \def\amqy.{\^a\d m} \def\Amqy.{\^A\d m}
 \def\amqf.{\=a\d m} \def\Amqf.{\=A\d m}
 \def\amqz.{\'a\d m} \def\Amqz.{\'A\d m}
\def\oav.{\v oa} \def\Oav.{\v Oa}
 \def\oah.{\`oa} \def\Oah.{\`Oa}
 \def\oay.{\^oa} \def\Oay.{\^Oa}
 \def\oaf.{\=oa} \def\Oaf.{\=Oa}
 \def\oaz.{\'oa} \def\Oaz.{\'Oa}
\def\oqa.{\d oa} \def\Oqa.{\d Oa}
 \def\oqav.{\d{\v o}a} \def\Oqav.{\d{\v O}a}
 \def\oqah.{\d{\`o}a} \def\Oqah.{\d{\`O}a}
 \def\oqay.{\d{\^o}a} \def\Oqay.{\d{\^O}a}
 \def\oqaf.{\d{\=o}a} \def\Oqah.{\d{\=O}a}
 \def\oqaz.{\d{\'o}a} \def\Oqaz.{\d{\'O}a}
\def\iav.{\v\i a} \def\Iav.{\v Ia}
 \def\iah.{\`\i a} \def\Iah.{\`Ia}
 \def\iay.{\^\i a} \def\Iay.{\^Ia}
 \def\iaf.{\=\i a} \def\Iaf.{\=Ia}
 \def\iaz.{\'\i a} \def\iaz.{\'\i a}
\def\iqa.{\d ia} \def\Iqa.{\d Ia}
 \def\iqav.{\d{\v\i}a} \def\Iqav.{\d{\v I}a}
 \def\iqah.{\d{\`\i}a} \def\Iqah.{\d{\`I}a}
 \def\iqay.{\d{\^\i}a} \def\Iqay.{\d{\^I}a}
 \def\iqaf.{\d{\=\i}a} \def\Iqaf.{\d{\=I}a}
 \def\iqaz.{\d{\'\i}a} \def\Iqaz.{\d{\'I}a}
\def\iaq.{i\d a} \def\Iaq.{I\d a}
 \def\iaqv.{\v\i\d a} \def\Iaqv.{\v I\d a}
 \def\iaqh.{\`\i\d a} \def\Iaqh.{\`I\d a}
 \def\iaqy.{\^\i\d a} \def\Iaqy.{\^I\d a}
 \def\iaqf.{\=\i\d a} \def\Iaqf.{\=I\d a}
 \def\iaqz.{\'\i\d a} \def\Iaqz.{\'I\d a}
\def\iqaq.{\d i\d a} \def\Iqaq.{\d I\d a}
 \def\iqaqv.{\d{\v\i}\d a} \def\Iqaqv.{\d{\v I}\d a}
 \def\iqaqh.{\d{\`\i}\d a} \def\Iqaqh.{\d{\`I}\d a}
 \def\iqaqy.{\d{\^\i}\d a} \def\Iqaqy.{\d{\^I}\d a}
 \def\iqaqf.{\d{\=\i}\d a} \def\Iqaqf.{\d{\=I}\d a}
 \def\iqaqz.{\d{\'\i}\d a} \def\Iqaqz.{\d{\'I}\d a}
\def\oev.{\v oe} \def\Oev.{\v Oe}
 \def\oeh.{\`oe} \def\Oeh.{\`Oe}
 \def\oey.{\^oe} \def\Oey.{\^Oe}
 \def\oef.{\=oe} \def\Oef.{\=Oe}
 \def\oez.{\'oe} \def\Oez.{\'Oe}
\def\oqe.{\d oe} \def\Oqe.{\d Oe}
 \def\oqev.{\d{\v o}e} \def\Oqev.{\d{\v O}e}
 \def\oqeh.{\d{\`o}e} \def\Oqeh.{\d{\`O}e}
 \def\oqey.{\d{\^o}e} \def\Oqey.{\d{\^O}e}
 \def\oqef.{\d{\=o}e} \def\Oqef.{\d{\=O}e}
 \def\oqez.{\d{\'o}e} \def\Oqez.{\d{\'O}e}
\def\uav.{\v ua} \def\Uav.{\v Ua}
 \def\uah.{\`ua} \def\Uah.{\`Ua}
 \def\uay.{\^ua} \def\Uay.{\^Ua}
 \def\uaf.{\=ua} \def\Uaf.{\=Ua}
 \def\uaz.{\'ua} \def\Uaz.{\'Ua}
\def\uqa.{\d ua} \def\Uqa.{\d Ua}
 \def\uqav.{\d{\v u}a} \def\Uqav.{\d{\v U}a}
 \def\uqah.{\d{\`u}a} \def\Uqah.{\d{\`U}a}
 \def\uqay.{\d{\^u}a} \def\Uqay.{\d{\^U}a}
```

```
654   \def\uqaf.{\d{\=u}a} \def\Uqaf.{\d{\=U}a}
655   \def\uqaz.{\d{\'u}a} \def\Uqaz.{\d{\'U}a}
656  \def\tq.{\d t} \def\Tq.{\d T}
657   \def\tqh.{\d th} \def\Tqh.{\d Th}
658   \def\dq.{\d d} \def\Dq.{\d D}
659   \def\dqh.{\d dh} \def\Dqh.{\d Dh}
660   \def\nq.{\d n} \def\Nq.{\d N}
661   \def\yq.{\d y} \def\Yq.{\d Y}
662   \def\sz.{\'s} \def\Sz.{\'S}
663   \def\lqq.{\d l} \def\Lq.{\d L}
664   \def\anqg.{a\d ng} \def\Anqg.{A\d ng}
665   \def\angq.{an\d g} \def\Angq.{An\d g}
666   \def\hq.{\d h} \def\Hq.{\d H}
667   \def\yz.{\'y} \def\Yz.{\'Y}
668   \def\yqb.{\d{\`y}} \def\Yqb.{\d{\`Y}}
669   \def\jhb.{\`\j h} \def\Jhb.{\`Jh}
670   \def\tqhb.{\d{\`t}h} \def\Tqhb.{\d{\`T}h}
671   \def\nb.{\`n} \def\Nb.{\`N}
672   \def\pb.{\`p} \def\Pb.{\`P}
673   \def\phb.{\`ph} \def\Phb.{\`Ph}
674   \def\bb.{\`b} \def\Bb.{\`B}
675   \def\mb.{\`m} \def\Mb.{\`M}
676   \def\iqa.{\d ia} \def\Iqa.{\d Ia}
677   \def\lb.{\`l} \def\Lb.{\`L}
678   \def\sb.{\`s} \def\Sb.{\`S}
679   \def\na.{na} \def\Na.{Na}
680   \def\yb.{\`y} \def\Yb.{\`Y}
681   \def\pq.{\d p} \def\Pq.{\d P}
682   \def\fq.{\d f} \def\Fq.{\d F}
683   \def\ax.{\.a} \def\Ax.{\.A}}
684 \def\latin{\def\Af.{\=A} \def\af.{\=a}
685   \def\Ef.{\=E} \def\ef.{\=e}
686   \def\If.{\=I} \def\if.{\=\i}
687   \def\Of.{\=O} \def\of.{\=o}
688  \def\aa.{\=a} \def\Aa.{\=A}
689  \def\ee.{\=e} \def\Ee.{\=E}
690  \def\ii.{\=\i} \def\Ii.{\=I}
691  \def\oo.{\=o} \def\Oo.{\=O}
692  \chardef\aE="1A
693   \chardef\oE="1B
694  \def\Aee.{\AE } \def\aee.{\aE }
695   \def\Oee.{\OE } \def\oee.{\oE }}
696 \def\lating{\def\[##1]{{\it ##1}} \def\(##1){{\bf ##1}} \def\und{{\eightrm~and~}}
697  \def\nm{{\eightit nom.}} \def\vo{{\eightit voc.}} \def\ac{{\eightit acc.}} \def\gn{{\eightit
gen.}}
698   \def\dat{{\eightit dat.}} \def\ab{{\eightit abl.}} \def\sg{{\eightit sg.}} \def\pl{{\eightit
pl.}}
699  \def\nmsg##1:{{\eightit nom.sg.} \(##1)} \def\vosg##1:{{\eightit voc.sg.} \(##1)}
700   \def\acsg##1:{{\eightit acc.sg.} \(##1)} \def\gnsg##1:{{\eightit gen.sg.} \(##1)}
701   \def\dtsg##1:{{\eightit dat.sg.} \(##1)} \def\absg##1:{{\eightit abl.sg.} \(##1)}
702   \def\nmpl##1:{{\eightit nom.pl.} \(##1)} \def\vopl##1:{{\eightit voc.pl.} \(##1)}
703   \def\acpl##1:{{\eightit acc.pl.} \(##1)} \def\gnpl##1:{{\eightit gen.pl.} \(##1)}
704   \def\dtpl##1:{{\eightit dat.pl.} \(##1)} \def\abpl##1:{{\eightit abl.pl.} \(##1)}
705  \def\dif[##1]{##1\[a], ##1\[am], ##1\[ae], ##1\[ae], ##1\[\af.],
706   ##1\(ae), ##1\(\af.s), ##1\(\af.rum), ##1\(\if.s), ##1\(\if.s)}
707  \def\diim[##1]{##1\[us], ##1\[um], ##1\[\if.], ##1\[\of.], ##1\[\of.],
708   ##1\(\if.), ##1\(\of.s), ##1\(\of.rum), ##1\(\if.s), ##1\(\if.s)}
709  \def\diima[##1]{\sg\nm\ ##1\[us], \vo\ ##1\[e], \ac\ ##1\[um], \gn\ ##1\[\if.], \dt\ ##1\[\of.],
710   \ab\ ##1\[\of.], \pl\nm\und\vo\ ##1\(\if.), \ac\ ##1\(\of.s), \gn\ ##1\(\of.rum), \dt\und\ab\
##1\(\if.s)}
711  \def\diin[##1]{##1\[um], ##1\[um], ##1\[\if.], ##1\[\of.], ##1\[\of.],
712   ##1\(a), ##1\(a), ##1\(\of.rum), ##1\(\if.s), ##1\(\if.s)}
713  \def\diif[##1]{##1\[er], ##1\[erum], ##1\[r\if.], ##1\[r\of.], ##1\[r\of.],
714   ##1\(r\if.), ##1\(r\of.s), ##1\(r\of.rum), ##1\(r\if.s), ##1\(r\if.s)}
715  \def\diiia:##1[##2]{\sg\nm\und\vo\ ##1, \ac\ ##2\[em], \gn\ ##2\[is], \dt\ ##2\[\if.],
\ab\ ##2\[e],
716   \pl\nm, \vo\und\ac\ ##2\[\ef.s], \gn\ ##2\[um], \dt\und\ab\ ##2\[ibus]}}
717 \def\math{\def\bap{\bigcap} \def\bcp{\bigcup}
718  \def\I{\hbox{I}}
719  \def\rar{\rightarrow}
720  \def\e:##1x##2[##3]{\hbox{$##1\times ##2^{##3}$}}
721   \def\C(##1,##2){{}^{##1}\hbox{C}_{##2}}
722   \def\Cc(##1,##2){\pmatrix{##1\cr##2}}
723   \def\det|##1|{\left\vert\matrix{##1}\right\vert}
724   \def\eno(##1){\eqno{(##1)}} \def\eln{\eqalign} \def\elno{\eqalignno}
725   \def\B[##1]{\hbox{##1}}
726   \def\mtx[##1]{\left[\matrix{##1}\right]}
727   \def\vtr[##1:##2:##3]{##1_{##2},\ldots,##1_{##3}}
728  \outer\def\plm ##1.##2\par{\medbreak
729   \noindent{\bf##1.\enspace}{\sl##2\par}
730   \ifdim\lastskip<\medskipamount \removelastskip\penalty55\medskip\fi}
731  \def\Ar{\bar{A}} \def\dr{\bar{d}} \def\fr{\bar{f}} \def\nr{\bar{n}} \def\ur{\bar{u}} \def\xr{\bar{x}}
```

```
 \def\yr{\bar{y}} \def\zr{\bar{z}}
\def\Ab{{\bf A}} \def\bb{{\bf b}} \def\Bb{{\bf B}} \def\Cb{{\bf C}} \def\Db{{\bf S}}
 \def\Eb{{\bf E}} \def\Ib{{\bf I}} \def\Pb{{\bf P}} \def\rb{{\bf r}} \def\Tb{{\bf T}}
 \def\ub{{\bf u}} \def\xb{{\bf x}} \def\yb{{\bf y}}
\def\xbh{\hat{\bf x}}
 \def\sd{\dot{s}} \def\Vd{\dot{V}} \def\xd{\dot{x}} \def\yd{\dot{y}} \def\zd{\dot{z}}
 \def\xbd{\dot{\xb}}
 \def\aht{\hat{a}} \def\bht{\hat{b}} \def\cht{\hat{c}}
 \def\thb{{\bf\theta}} \def\thh{\hat{\theta}}
 \def\thbh{\hat{\bf\theta}} \def\xibh{\hat{\bf\xi}}
\def\Nm{{\rm N}} \def\Tm{{\rm T}} \def\Zm{{\rm Z}}
\def\fan{{\rm f}}
\def\p{\prime} \def\pp{{\prime\prime}} \def\ppp{{\prime\prime\prime}}
\def\alp{\aleph}
 \def\and{\hbox{and}}
 \def\aph{\alpha} \def\Aph{\Alpha}
 \def\apx{\approx}
 \def\bdbl{\begindoublecolumns}
 \def\bta{\beta}
 \def\bo{\cal O}
 \def\cdt{\cdot} \def\cds{\cdots}
 \def\cir{\circ}
 \def\dg{\hbox{$^\circ$}}
 \def\dlt{\delta} \def\Dlt{\Delta}
 \def\edbl{\enddoublecolumns}
 \def\eightp{\eightpoint}
 \def\ems{\emptyset}
 \def\epn{\epsilon} \def\Epn{\Epsilon}
 \def\fll{\forall}
 \def\fnt{\footnote}
 \def\gmm{\gamma} \def\Gmm{\Gamma}
 \def\iny{\infty}
 \def\itg{\intop\nolimits}
 \def\ity{\infty}
 \def\kpa{\kappa}
 \def\lds{\ldots}
 \def\llr{\longleftrightarrow} \def\Llr{\Longleftrightarrow}
 \def\lmd{\lambda} \def\Lmd{\Lambda}
 \def\lng{\langle}
 \def\lra{\leftrightarrow} \def\Lra{\Leftrightarrow}
 \def\lrw{\longrightarrow} \def\Lrw{\Longrightarrow}
 \def\nind{\noindent}
 \def\ninep{\ninepoint}
 \def\ols{\obeylines}
 \def\opl{\oplus}
 \def\ovr{\over}
 \def\omg{\omega} \def\Omg{\Omega}
 \def\oss{\obeyspaces}
 \def\pint{\parindent}
 \def\prd{\prod} \def\ptl{\partial}
 \def\prp{\propto}
 \def\qiv{\equiv}
 \def\rar{\rightarrow}
 \def\rng{\rangle}
 \def\sgm{\sigma} \def\Sgm{\Sigma}
 \def\stt{\hbox{ s.t. }}
 \def\sub{\subset}
 \def\tms{\times}
 \def\tta{\theta} \def\Tta{\Theta}
 \def\vep{\varepsilon}
 \def\vph{\varphi}
 \def\vrt{\vert}
 \def\xst{\exists}
\def\q{\quad} \def\qq{\qquad}
\def\lf{\left} \def\rt{\right}
\def\Abs{\mathop{\rm abs}\nolimits}
 \def\angle{\mathop{\rm angle}\nolimits}
 \def\arc{\mathop{\rm arc}\nolimits}
 \def\cosec{\mathop{\rm cosec}\nolimits}
 \def\D{{\rm d}}
 \def\Div{\mathop{\rm div}\nolimits}
 \def\E{\mathop{\rm E}\nolimits}
 \def\Grad{\mathop{\rm grad}\nolimits}
 \def\Kur{\mathop{\cal K}\nolimits}
 \def\Min{\mathop{\rm min}\nolimits}
 \def\Max{\mathop{\rm max}\nolimits}
 \def\mod{\mathop{\rm mod}\nolimits}
 \def\Mom{\mathop{\cal M}\nolimits}
 \def\Med{\mathop{\rm med}\nolimits} \let\med=\Med
 \def\Mad{\mathop{\rm mad}\nolimits} \let\mad=\Mad
 \def\Neb{\mathop{\aleph}\nolimits}
 \def\Nor{\mathop{{\cal N}^c_v}\nolimits}
```

```
  \def\Oc{\mathop{\rm O}\nolimits}
  \def\Per{\mathop{\cal P}\nolimits}
  \def\Rl{\mathop{\rm Re}\nolimits}
  \def\sgn{\mathop{\rm sgn}\nolimits}
  \def\Vet{\mathop{\cal V}\nolimits}
 \def\centre{\centerline}
  \def\haln{\halign}
  \def\ovln{\overline}
  \def\lline{\leftline}
  \def\mspn{\multispan}
  \def\pshp{\parshape}
  \def\rline{\rightline}
  \def\sskp{\smallskip}
 \def\nxs{{\raise1pt\hbox{$/$}\kern-5.1pt\xst}
 \def\proof{\noindent{\bf Proof.:} }
 \def\endprf{\hfill$\sqr$}
 \def\qedprf{\par\hfill{q.e.d.}}}
\def\nihongo{\def\kanji(##1)[##2]{\setbox1=\hbox{##1} \dtmp=\wd1
  \setbox2=\vbox{\hsize=\dtmp\noindent\sevenrm--\hfil##2\hfil--}
  \hbox{}\kern-1em$\mathrel{\mathop{\kern\z@\copy1}\limits_{\sevenrm\kern.5em\raise.5em\copy2}}$}
 \def\radcal(##1)[##2]{\setbox1=\hbox{##1} \dtmp=\wd1
  \setbox2=\vbox{\hsize=\dtmp\noindent\sevenbf--\hfil##2\hfil--}
  \hbox{}\kern-1em$\mathrel{\mathop{\kern\z@\copy1}\limits_{\sevenrm\kern.5em\raise.5em\copy2}}$}
 \def\Ou.{\=O} \def\ou.{\=o}
 \def\Uu.{\=U} \def\uu.{\=u}
 \def\Arak.{\kanji(Ara)[k]}
 \def\Hiroshis.{\kanji(Hiroshi)[s]}
 \def\kawa.{\radcal(kawa)[b]} \def\Kawa.{\radcal(Kawa)[b]}
 \def\Suzug.{\kanji(Suzu)[g]}
}
\def\norge{\def\aA{\accent23a}
  \def\axx.{\aA } \def\Axx.{\AA }
 \def\oz.{\'o} \def\Oz.{\'O}
  \chardef\oO="1C
  \def\ozz.{\oO } \def\Ozz.{\O }}
\def\pali{\def\aa.{\=a} \def\Aa.{\=A}
  \def\ii.{\=\i} \def\Ii.{\=I}
  \def\uu.{\=u} \def\Uu.{\=U}
 \def\dq.{\d d} \def\Dq.{\d D}
  \def\dqh.{\d dh} \def\Dqh.{\d Dh}
  \def\tq.{\d t} \def\Tq.{\d T}
  \def\tqh.{\d th} \def\Tqh.{\d Th}
  \def\nq.{\d n} \def\Nq.{\d N}
  \def\nl.{\~n} \def\Nl.{\~N}
 \def\nx.{\.n} \def\Nx.{\.N}}
\def\physics{\def\unt[##1]{$\rm ##1$}
}
\def\polish{\def\aj.{\c a} \def\Aj.{\c A}
 \def\cz.{\'c} \def\Cz.{\'C}
 \def\ej.{\c e} \def\Ej.{\c E}
 \def\lz.{\'l} \def\Lz.{\'L}
 \def\lzz.{\l } \def\Lzz.{\L }
 \def\nz.{\'n} \def\Nz.{\'N}
 \def\oz.{\'o} \def\Oz.{\'O}
 \def\sz.{\'s} \def\Sz.{\'S}
 \def\zz.{\'z} \def\Zz.{\'Z}
 \def\zx.{\.z} \def\Zx.{\.Z}
 \def\dzz.{d\'z} \def\Dzz.{D\'z}
 \def\dzx.{d\.z} \def\Dzx.{D\.z}}
\def\russian{\def\iv.{\v\i} \def\Iv.{\v I} %
 \def\ee.{\"e} \def\Ee.{\"E}}
\def\sanskrit{\def\aa.{\=a} \def\Aa.{\=A}
  \def\ii.{\=\i} \def\Ii.{\=I}
  \def\uu.{\=u} \def\Uu.{\=U}
 \def\dq.{\d d} \def\Dq.{\d D}
  \def\dqh.{\d dh} \def\Dqh.{\d Dh}
  \def\tq.{\d t} \def\Tq.{\d T}
  \def\tqh.{\d th} \def\Tqh.{\d Th}
 \def\nx.{\.n} \def\Nx.{\.N}
  \def\nl.{\~n} \def\Nl.{\~N}
 \def\rq.{\d r} \def\Rq.{\d R}
  \def\rqh.{\d rh} \def\Rqh.{\d Rh}
  \def\lqq.{\d l} \def\Lq.{\d L}
  \def\mq.{\d m} \def\Mq.{\d M}
  \def\hq.{\d h} \def\Hq.{\d H}
 \def\sz.{\'s} \def\Sz.{\'S}
  \def\sq.{\d s} \def\Sq.{\d S}}
\def\serbo{\def\cv.{\v c} \def\Cv.{\v C}
 \def\cz.{\'c} \def\Cz.{\'C}
 \def\dz.{\'d} \def\Dz.{\'D}
 \def\dzv.{d\v z} \def\Dzv.{D\v z}
 \def\sv.{\v s} \def\Sv.{\v S}
```

```
 \def\zv.{\v z} \def\Zv.{\v Z}}
\def\slovak{\def\az.{\'a} \def\Az.{\'A}
 \def\iz.{\'\i} \def\Iz.{\'I}
 \def\yz.{\'y} \def\Yz.{\'Y}
 \def\oz.{\'o} \def\Oz.{\'O}
 \def\uz.{\'u} \def\Uz.{\'U}
 \def\ae.{\"a} \def\Ae.{\"A}
 \def\oy.{\^o} \def\Oy.{\^O}
 \def\rz.{\'r} \def\Rz.{\'R}
 \def\lz.{\'l} \def\Lz.{\'L}
 \def\cv.{\v c} \def\Cv.{\v C}
 \def\sv.{\v s} \def\Sv.{\v S}
 \def\zv.{\v z} \def\Zv.{\v Z}
 \def\dzv.{d\v z} \def\Dzv.{D\v z}
 \def\dv.{\v d} \def\Dv.{\v D}
 \def\tv.{\v t} \def\Tv.{\v T}
 \def\nv.{\v n} \def\Nv.{\v N}
 \def\lv.{\v l} \def\Lv.{\v L}}
\def\tex{\long\def\veb##1{\def\next{##1}{\tt\frenchspacing\expandafter\strip\meaning\next}}
 \def\strip##1>{}
 \def\bx{{\it bx}}    % TeX book
 \def\kx{{\it kx}}    % KiteX
 \def\mx{{\it mx}}    % manmac
 \def\plx{{\it plx}}  % plain TeX
 \def\px{{\it px}}    % primitive TeX
}
\def\vietnamese{\def\dt.{d\kern-.5em\raise.25em\hbox{--}}
  \def\Dt.{{D\kern-.75em\raise.1em\hbox{--}\kern.25em}}
 \def\ap.{\u a} \def\Ap.{\u A}
 \def\ay.{\^a} \def\Ay.{\^A}
 \def\ey.{\^e} \def\Ey.{\^E}
 \def\oy.{\^o} \def\Oy.{\^O}
 \def\az.{\'a} \def\Az.{\'A}
 \def\as.{a\kern-.5em\raise.4em\hbox{?}} \def\As.{{A\kern-.6em\raise.59em\hbox{?}\kern.1em}}
 \def\al.{\~a} \def\Al.{\~A}
 \def\aq.{\d a} \def\Aq.{\d A}
 \def\ez.{\'e} \def\Ez.{\'E}
 \def\eh.{\`e} \def\Eh.{\`E}
 \def\es.{e\kern-.5em\raise.39em\hbox{?}} \def\Es.{E\kern-.5em\raise.62em\hbox{?}}
 \def\el.{\~e} \def\El.{\~E}
 \def\eq.{\d e} \def\Eq.{\d E}}
\def\zhongwen{\def\af.{\=a} \def\az.{\'a} \def\av.{\v a} \def\ah.{\`a}
  \def\if.{\=\i} \def\iz.{\'\i} \def\iv.{\v\i} \def\ih.{\`\i}
  \def\of.{\=o} \def\oz.{\'o} \def\ov.{\v o} \def\oh.{\`o}
  \def\ouf.{\=ou} \def\ouz.{\'ou} \def\ouv.{\v ou} \def\ouh.{\`ou}
  \def\uf.{\=u} \def\uz.{\'u} \def\uv.{\v u} \def\uh.{\`u}
  \def\uof.{u\=o} \def\uoz.{u\'o} \def\uov.{u\v o} \def\uoh.{u\`o}
 \def\putji(##1)[##2]{\begingroup\setbox1=\hbox{##1} \dtmp=\wd1
  \setbox2=\vbox{\hsize=\dtmp\noindent\sevenrm--\hfil##2\hfil--}
  \hbox{}\kern-1em$\mathrel{\mathop{\kern\z@\copy1}\limits_{\sevenrm\kern.5em\raise.5em\copy2}}$\endgroup}
 \def\putbs(##1){\begingroup\setbox1=\hbox{##1} \dtmp=\wd1
  \setbox2=\vbox{\hsize=\dtmp\noindent\sevenrm$\cdot$\hfil} \hbox{}\kern-1em
  $\mathrel{\mathop{\kern\z@\copy1}\limits_{\sevenrm\kern.5em\raise.7em\copy2}}$\lineskip-.5em\endgroup}
 \def\aifc.{\putji(\=ai)[c]} \def\Aifc.{\putji(\=Ai)[c]} \def\aifcm.{{\it grief or sorrow}}
 \def\baofc.{\putji(b\=ao)[c]} \def\Baofc.{\putji(B\=ao)[c]} \def\baofcm.{{\it praise or
commend}}
  \def\biaovc.{\putji(bi\v ao)[c]} \def\Biaovc.{\putji(Bi\v ao)[c]} \def\biaovcm.{{\it surface}}
 \def\caizc.{\putji(c\'ai)[c]} \def\Caizc.{\putji(C\'ai)[c]} \def\caizcm.{{\it cut paper
or cloth}}
 \def\chazng.{\putbs(ch\'ang)} \def\Chazng.{\putbs(Ch\'ang)}
 \def\daihc.{\putji(d\`ai)[c]} \def\Daihc.{\putji(D\`ai)[c]} \def\daihcm.{{\it a sack or
bag}}
 \def\gofng.{\putbs(g\=ong)} \def\Gofng.{\putbs(G\=ong)}
  \def\guovc.{\putji(gu\v o)[c]} \def\Guovc.{\putji(Gu\v o)[c]} \def\guovcm.{{\it wrap or
bind}}
 \def\livc.{\putji(l\v\i)[c]} \def\Livc.{\putji(L\v\i)[c]} \def\livcm.{{\it lining or inside}}
  \def\lozng.{\putbs(l\'ong)} \def\Lozng.{\putbs(L\'ong)}
 \def\miznгry.{\putji(m\'\i ng)[ry]} \def\Miznгry.{\putji(M\'\i ng)[ry]}
 \def\niaovc.{\putji(ni\v ao)[c]} \def\Niaovc.{\putji(Ni\v ao)[c]} \def\niaovcm.{{\it slender
and delicate}}
 \def\qiuzc.{\putji(qi\'u)[c]} \def\Qiuzc.{\putji(Qi\'u)[c]} \def\qiuzcm.{{\it fur coat}}
 \def\rih.{\putbs(r\`\i)} \def\Rih.{\putbs(R\`\i)} \def\rihm.{{\it the Sun}}
 \def\shuaifc.{\putji(shu\=ai)[c]} \def\Shuaifc.{\putji(Shu\=ai)[c]} \def\shuaifcm.{{\it
decline}}
 \def\xiafngc.{\putji(xi\=ang)[c]} \def\Xiafngc.{\putji(Xi\=ang)[c]} \def\xiafngcm.{{\it
assist}}
  \def\xiehc.{\putji(xi\`e)[c]} \def\Xiehc.{\putji(Xi\`e)[c]} \def\xiehcm.{{\it blaspheme}}
  \def\xizc.{\putji(x\'\i)[c]} \def\Xizc.{\putji(X\'\i)[c]} \def\xizcm.{{\it raid}}
 \def\yif.{\putbs(y\=i)} \def\Yif.{\putbs(Y\=i)} \def\yifm.{{\it clothes}}
  \def\yihc.{\putji(y\`\i)[c]} \def\Yihc.{\putji(Y\`\i)[c]} \def\yihcm.{{\it descendants}}
  \def\yueh.{\putbs(yu\`e)} \def\Yueh.{\putbs(Yu\`e)} \def\yuehm.{{\it the moon}}
 \def\zhafngg.{\putji(zh\=ang)[g]} \def\Zhafngg.{\putji(Zh\=ang)[g]}
```

```
971   \def\zhihc.{\putji(zh\`\i)[c]} \def\Zhihc.{\putji(Zh\`\i)[c]} \def\zhihcm.{{\it control,
make, or system}}
972   \def\zhofngc.{\putji(zh\=ong)[c]} \def\Zhofngc.{\putji(Zh\=ong)[c]} \def\zhofngcm.{{\it
inner feelings}}
973   \def\zhuafngc.{\putji(zhu\=ang)[c]} \def\Zhuafngc.{\putji(Zhu\=ang)[c]} \def\zhuafngcm.{{\it
clothing}}
974 }
```

§ A.13 Degenerate test and other programs

```
% degen.m
Dim=40; Xbunch=(0:2:Dim)'; Ycore=ones(size(Xbunch)); X=[]; Y=[];
for j=0:2:Dim,
  X=[X;Xbunch]; Y=[Y;(j*Ycore)];
end
[Xsq,Ysq]=voronoi(X,Y); Tri=delaunay(X,Y); NumTri=size(Tri,1); [XHex,YHex]=voronoi(X,Y);
% honey.m
Sq3=sqrt(3); Dim=40; Shift=1; y=1:Sq3:10; X=[]; Y=[];
Xinit=0; Xbunch=(Shift:6:Dim)'; Ycore=ones(size(Xbunch));
for j=0:(2*Sq3):Dim,
  X=[X;Xbunch]; Y=[Y;j*Ycore];
end
Xinit=3; Xbunch=(3:6:Dim)'; Ycore=ones(size(Xbunch));
for j=Sq3:(2*Sq3):Dim,
  X=[X;Xbunch]; Y=[Y;j*Ycore];
end
Tri=delaunay(X,Y); NumTri=size(Tri,1); [XHex,YHex]=voronoi(X,Y);
% cover.m
Sq3=sqrt(3); Dim=40; y=1:Sq3:10; X=[]; Y=[]; Xinit=0;
Xbunch=(0:6:Dim)'; Ycore=ones(size(Xbunch));
for j=0:(2*Sq3):Dim,
  X=[X;Xbunch]; Y=[Y;j*Ycore];
end
Xinit=3; Xbunch=(3:6:Dim)'; Ycore=ones(size(Xbunch));
for j=Sq3:(2*Sq3):Dim,
  X=[X;Xbunch]; Y=[Y;j*Ycore];
end
[XHex,YHex]=voronoi(X,Y); NumE=size(XHex,2); NeighE=sparse(NumE,NumE);
for i=1:(NumE-1),
  for j=(i+1):NumE,
    if(((XHex(1,i)==XHex(1,j)) & (YHex(1,i)==YHex(1,j))) | ...
     ((XHex(1,i)==XHex(2,j)) & (YHex(1,i)==YHex(2,j))) | ...
     ((XHex(2,i)==XHex(1,j)) & (YHex(2,i)==YHex(1,j))) | ...
     ((XHex(2,i)==XHex(2,j)) & (YHex(2,i)==YHex(2,j))))
      NeighE(i,j)=1; NeighE(j,i)=1;
    end
  end
end
% covers.m
Xmid=[]; Ymid=[];
for i=1:NumE,
  Xmid=[Xmid;((XHex(1,i)+XHex(2,i))/2)]; Ymid=[Ymid;((YHex(1,i)+YHex(2,i))/2)];
end
[I,J]=find(tril(NeighE)); NumV=size(I,1); Xcov=[]; Ycov=[];
for i=1:NumV,
  Xcov=[Xcov;[Xmid(I(i,1),1),Xmid(J(i,1),1)]]; Ycov=[Ycov;[Ymid(I(i,1),1),Ymid(J(i,1),1)]];
end
Xcovt=Xcov'; Ycovt=Ycov'; NumE1=size(Xcov,1); NeighE1=sparse(NumE1,NumE1);
for i=1:(NumE1-1),
  for j=(i+1):NumE1,
    if(((Xcovt(1,i)'==Xcovt(1,j)') & (Ycovt(1,i)'==Ycovt(1,j)')) | ...
     ((Xcovt(1,i)'==Xcovt(2,j)') & (Ycovt(1,i)'==Ycovt(2,j)')) | ...
     ((Xcovt(2,i)'==Xcovt(1,j)') & (Ycovt(2,i)'==Ycovt(1,j)')) | ...
     ((Xcovt(2,i)'==Xcovt(2,j)') & (Ycovt(2,i)'==Ycovt(2,j)')))
      NeighE1(i,j)=1; NeighE1(j,i)=1;
    end
  end
end
Xmid=[]; Ymid=[];
for i=1:NumE1,
  Xmid=[Xmid;((Xcovt(1,i)'+Xcovt(2,i)')/2)]; Ymid=[Ymid;((Ycovt(1,i)'+Ycovt(2,i)')/2)];
end
[I,J]=find(tril(NeighE1)); NumV=size(I,1); Xc1=[]; Yc1=[];
for i=1:NumV,
  Xc1=[Xc1;[Xmid(I(i,1),1),Xmid(J(i,1),1)]]; Yc1=[Yc1;[Ymid(I(i,1),1),Ymid(J(i,1),1)]];
end
% coverss.mXc1t=Xc1'; Yc1t=Yc1';
NumE2=size(Xc1,1); NeighE2=sparse(NumE2,NumE2);
for i=1:(NumE2-1),
  for j=(i+1):NumE2,
```

```
    if(((Xc1t(1,i)'==Xc1t(1,j)') & (Yc1t(1,i)'==Yc1t(1,j)')) | ...
     ((Xc1t(1,i)'==Xc1t(2,j)') & (Yc1t(1,i)'==Yc1t(2,j)')) | ...
     ((Xc1t(2,i)'==Xc1t(1,j)') & (Yc1t(2,i)'==Yc1t(1,j)')) | ...
     ((Xc1t(2,i)'==Xc1t(2,j)') & (Yc1t(2,i)'==Yc1t(2,j)')))
      NeighE2(i,j)=1; NeighE2(j,i)=1;
    end
  end
end
Xmid=[]; Ymid=[];
for i=1:NumE2,
  Xmid=[Xmid;((Xc1t(1,i)'+Xc1t(2,i)')/2)]; Ymid=[Ymid;((Yc1t(1,i)'+Yc1t(2,i)')/2)];
end
[I,J]=find(tril(NeighE2)); NumV=size(I,1); Xc2=[]; Yc2=[];
for i=1:NumV,
  Xc2=[Xc2;[Xmid(I(i,1),1),Xmid(J(i,1),1)]]; Yc2=[Yc2;[Ymid(I(i,1),1),Ymid(J(i,1),1)]];
end
% crop.m     by K N J Tiyapan, 15 July 2001
clear all; format long g; format compact; NumCell=1000; rand('state',sum(100*clock));
[XVoro,YVoro]=voronoi(X,Y); SizeV=size(XVoro,2); Xv=[]; Yv=[];
for j=1:SizeV,
  if((((XVoro(1,j)>0) & (XVoro(1,j)<1)) & ((YVoro(1,j)>0) & (YVoro(1,j)<1))) | ...
   (((XVoro(2,j)>0) & (XVoro(2,j)<1)) & ((YVoro(2,j)>0) & (YVoro(2,j)<1))))
    Xv=[Xv,XVoro(:,j)]; Yv=[Yv,YVoro(:,j)];
  end
end
```

§ A.14 Face statistics in n dimensions

```
% statsgenn.m    by K N J Tiyapan, 1st July, 2001
echo off; clear all; format short g; more off;
pt1 =fopen('./v50.dat','r'); sc1 =fscanf(pt1, '%d', 4);
Dimension=sc1(1,1);  NumVAll =sc1(2,1); NumC =sc1(3,1);
sc2 =fscanf(pt1, '%f', [Dimension, NumVAll]); VerticeAll =sc2';
CVMat =sparse(NumC, NumVAll); CFrame =ones(NumC, 1); VCFrame=zeros(NumVAll,1);
VFrame=ones(NumVAll,1);
for i=1:NumC,
  sc1 =fscanf(pt1, '%d', 1);
  for j=1:sc1,
    sc2 =fscanf(pt1, '%d', 1); Num =sc2+1; CVMat(i,Num) =1;
    if ( max(abs(VerticeAll(Num, :))) > 0.5 )
      CFrame(i,1) =0; VFrame(Num,1)=0;
    end
  end
end
fclose(pt1);
for i=1:NumC,
  VInC=find(CVMat(i,:)'); NumVInC=size(VInC,1);
  if(CFrame(i,1)==1)
    for j=1:NumVInC,
      VCFrame(VInC(j,1),1)=1;
    end
  end
end
CVNiceCMat=[];
for i=1:NumC,
  if(CFrame(i,1)==1)
    CVNiceCMat=[CVNiceCMat;CVMat(i,:)];
  end
end
CNumVNiceCMat=sum(CVNiceCMat,2); NumV=sum(VCFrame);
VVCFrameMat=zeros(NumV,2); Vertice=zeros(NumV,Dimension); Count=0;
for i=1:NumVAll,
  if(VCFrame(i,1)==1)
    Count=Count+1;
    VVCFrameMat(Count,1)=i; VVCFrameMat(Count,2)=Count;
    Vertice(Count,:)=VerticeAll(i,:);
  end
end
pt3=fopen('./n50.dat','w'); pt2=fopen('./c50.dat','r'); line=fgetl(pt2);
sc1 =fscanf(pt2, '%d', 1); sc2=fscanf(pt2,'%f',[Dimension,NumC]); Cell=sc2';
fclose(pt2); CNeighCCMat=sparse(NumC, NumC); t=cputime; FVAllMat=[]; FNumVAllMat=[];
for i=1:(NumC-1),
  for j=(i+1):NumC,
    VShared=and(CVMat(i,:), CVMat(j,:)); NumShared =sum(VShared, 2);
    NumFVAllMat=size(FVAllMat,1);
    if (NumShared >= Dimension)
      CNeighCCMat(i,j) =1; CNeighCCMat(j,i) =1; Exist=0;
      for k=1:NumFVAllMat,
        MatchExistingFV=sum(and(VShared,FVAllMat(k,:)),2);
        if(MatchExistingFV>=Dimension)
```